KOMPUTASI SEMANTIK & AI

Transformasi Pengetahuan menuju Kecerdasan



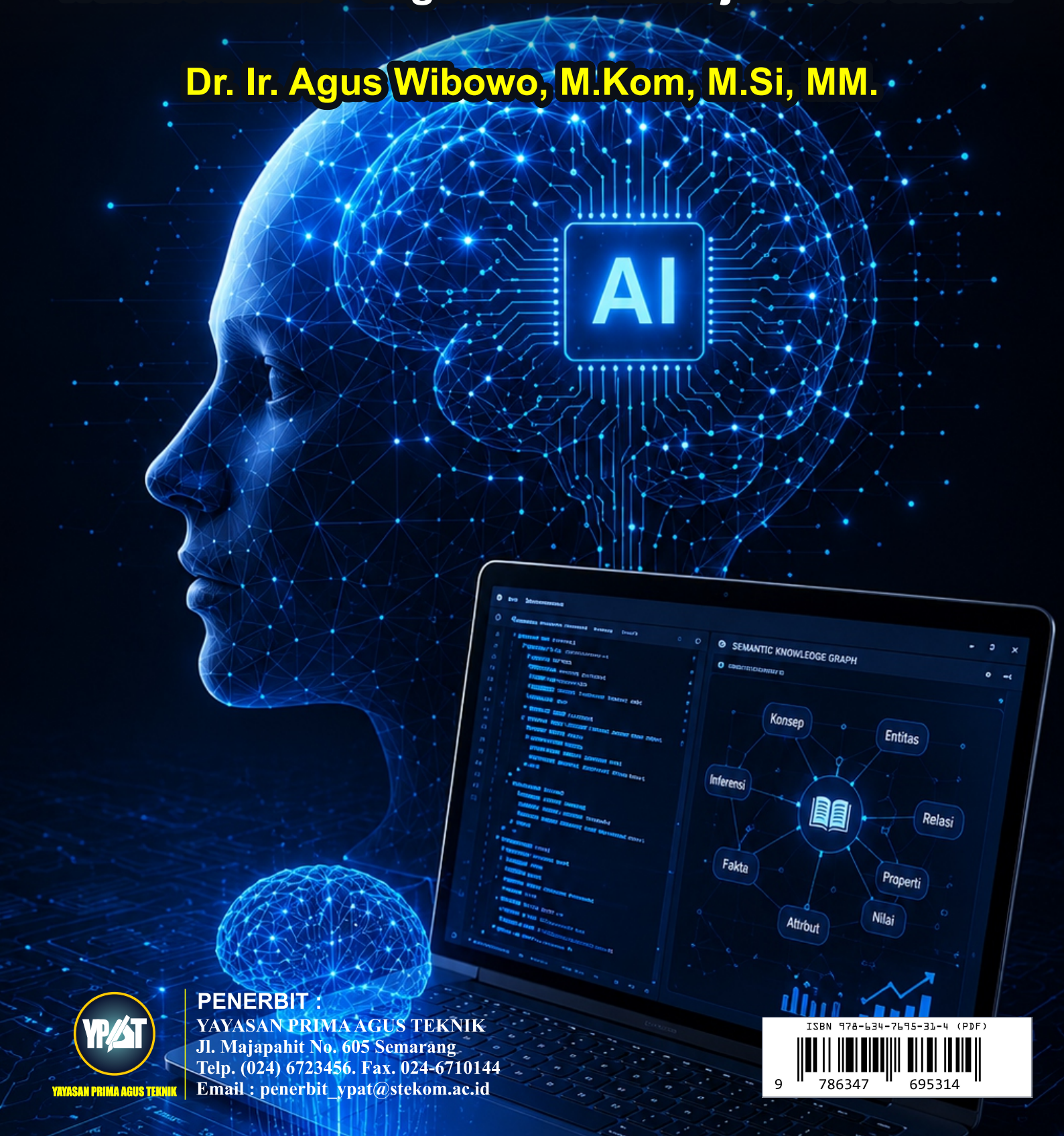
Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

YAYASAN PRIMA AGUS TEKNIK

KOMPUTASI SEMANTIK & AI

Transformasi Pengetahuan menuju Kecerdasan

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-31-4 (PDF)
9 786347 695314

KOMPUTASI SEMANTIK & AI :
Transformasi Pengetahuan menuju Kecerdasan

Penulis :
Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

ISBN : 978-634-7695-31-4 (PDF)

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa ijin dari penulis

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat-Nya sehingga buku "*Komputasi Semantik & AI: Transformasi Pengetahuan menuju Kecerdasan*" ini dapat terselesaikan dengan sempurna. Perkembangan pesat dalam bidang kecerdasan buatan (AI) dan komputasi semantik selama dekade terakhir telah menghasilkan lompatan konseptual dan teknis yang signifikan. Kemampuan sistem untuk memahami, merepresentasikan, dan memanfaatkan pengetahuan telah mengalami transformasi dari sekadar model statistik ke sistem hibrida yang menggabungkan representasi simbolik, struktur graf-pengetahuan, dan kemampuan pembelajaran mendalam yang kuat. Buku ini disusun dengan tujuan menghadirkan telaah yang menyeluruh dan terintegrasi mengenai perkembangan tersebut—menggabungkan fondasi teoretis, algoritma, eksperimen, serta aplikasi lintas domain.

Kebutuhan akan narasi yang menjembatani teori dan praktik menjadi dasar penyusunan buku ini. Dalam praktik riset maupun pembangunan sistem, sering terdapat jurang antara konsep abstrak (seperti ontologi, logika deskriptif, atau representasi semantik) dan implementasi efektif pada skala nyata (misalnya, deployment LLM, pipeline ekstraksi pengetahuan pada data medis, atau sistem deteksi fraud berbasis graf). Buku ini dirancang untuk mengatasi jurang tersebut dengan menyajikan: penjelasan konseptual yang matang, formulasi matematis yang diperlukan, deskripsi algoritma dan arsitektur, serta hasil eksperimen yang dapat menjadi referensi bagi replikasi dan pengembangan lanjutan.

Buku ini tersusun atas empat belas bab yang masing-masing menyajikan topik inti secara mendetail. Berikut ringkasan tiap bagian utama beserta tujuan pedagogis dan teknisnya, Bab 1 Bab pembuka memetakan evolusi model bahasa besar (LLM), termasuk fondasi teknis jaringan saraf modern, proses pembangunan model bahasa berskala besar, serta pergeseran paradigma yang dipicu oleh arsitektur transformer. Selain aspek teknis, bab ini menyoroti tantangan akuntabilitas, bias, dan isu keselamatan yang muncul ketika kemampuan model meningkat. Tujuan bab ini adalah memberikan konteks historis dan kritis sehingga pembaca mampu mengevaluasi kemajuan LLM secara seimbang.

Bab 2 ini memperkenalkan kerangka komputasi semantik sebagai mekanisme integrasi antara representasi masalah dan penawaran solusi. Terminologi dan bahasa formal—seperti SCDL (*Solution Component Description Language*), SQDL (*Solution Query Description Language*), dan SNL—dijabarkan secara sistematis. Bab ini juga mengulas mekanisme pencocokan, evaluasi komposisi layanan, dan cakupan aplikasi praktisnya dalam ekosistem pertukaran solusi. Bab 3 menyajikan metode pembobotan istilah menggunakan pendekatan random walk, bab ini menjelaskan algoritma yang diusulkan, desain eksperimen, serta evaluasi performa pada tugas klasifikasi teks. Diskusi meliputi parameterisasi random walk, konstruk graf istilah, dan analisis sensitivitas terhadap data beragam.

Bab 4 fokus pada penalaran terdistribusi pada ontologi web yang saling terhubung, bab ini menyajikan model distribusi resolusi logis, adaptasi untuk logika deskriptif seperti ALC, serta evaluasi metode resolusi terdistribusi. Pembahasan juga mencakup tantangan

konsistensi, kompleksitas komputasi, dan skenario aplikasi. Bab 5 menguraikan teknik segmentasi dan deteksi adegan pada video dengan pendekatan multimodal yang memanfaatkan struktur shot, fitur visual, dan informasi temporal. Bab ini mencakup metode pengelompokan sekuensial optimal, kompleksitas komputasi, serta eksperimen pada dataset benchmark. Bab 6 menyajikan penggunaan knowledge graph dan XAI untuk analisis proses bisnis, termasuk permodelan pohon keputusan stokastik, justifikasi desain tujuan, dan integrasi knowledge graph dengan machine learning untuk menciptakan sistem XAI-PA (*Explainable AI for Process Analysis*). Bab ini memfokuskan pada bagaimana representasi pengetahuan meningkatkan keterterangan dan akurasi analitik proses.

Bab 7 memberikan tinjauan komprehensif tentang evolusi dari machine learning ke deep learning, arsitektur utama, algoritma pelatihan, kerangka kerja perangkat lunak, serta keterkaitan dengan big data. Bab ini juga membahas hambatan teknis dan riset yang masih terbuka. Bab 8 membahas perkembangan CNN, evolusi metode pengenalan visual, arsitektur utama, serta evaluasi komparatif ekstraksi fitur dan fusi pada benchmark gambar. Disertakan studi kasus sistem pengenalan konsep visual berbasis fusi fitur. Bab 9 menguraikan penerapan deep learning dan *Neural Architecture Search* (NAS) untuk solusi real-time industri, mekanisme evolusi grafik (NEAT dan CA), knowledge distillation, serta studi kasus benchmark untuk permainan 2D dan pelacakan gerak trailer.

Bab 10 memaparkan dasar teori fusi multimodal, isu-isu kritis, teknik reduksi dimensi dan fusi fitur berbasis kernel, serta metodologi berbasis teori entropi dan penerapannya dalam pengenalan emosi audio. Bab 11 menyajikan kerangka ensemble deep learning untuk deteksi peristiwa video, protokol evaluasi, dan analisis performa ensemble dibandingkan pendekatan tunggal. Bab 12 menguraikan adaptive *Deep Belief Network* (DBN) untuk ekstraksi pengetahuan pada data medis, metodologi adaptasi, dan evaluasi pada dataset klinis. Bab ini menekankan isu privasi, validitas klinis, serta prospek translasi ke praktik medis. Bab 13 membahas pemanfaatan komputasi graf untuk deteksi kejahatan finansial, karakteristik dan tren penipuan terkini, kendala teknis, serta arah riset untuk meningkatkan ketahanan sistem deteksi. Bab 14 menampilkan metodologi pembuatan judul otomatis untuk sumber belajar daring menggunakan transformer pretrained, termasuk fine-tuning GPT-2, evaluasi sistem peringkasan, dan pembuatan judul hierarkis untuk jalur pembelajaran.

Setiap bab dikembangkan menurut kerangka yang konsisten: landasan teoritis, formulasi metodologis dan matematis, deskripsi arsitektur dan algoritma, protokol eksperimen, hasil evaluasi, serta diskusi kritis yang menyoroti keterbatasan dan peluang pengembangan. Pendekatan ini bertujuan memfasilitasi pemahaman mendalam sekaligus kemampuan replikasi ilmiah. Penjelasan teknis dilengkapi dengan rujukan yang relevan pada literatur terkini sehingga pembaca dapat menelusuri sumber primer untuk kajian lebih lanjut.

Buku ini ditujukan kepada beragam kalangan profesional dan akademik: peneliti dan dosen yang membutuhkan rujukan teoretis dan eksperimental; mahasiswa pascasarjana yang mencari panduan untuk replikasi dan pengembangan penelitian; praktisi industri yang merancang sistem cerdas terapan; serta pembuat kebijakan yang memerlukan pemahaman komprehensif mengenai implikasi teknis dan etis AI. Meskipun beberapa bagian mengandung

notasi matematis dan rincian implementasi, penyajian dilakukan sedemikian rupa untuk menjaga keterbacaan tanpa mengorbankan ketelitian ilmiah.

Dengan memadukan kajian teoritis, metode eksperimen, dan aplikasi lintas domain, buku ini diharapkan berkontribusi pada beberapa hal berikut, menjadi sumber rujukan akademik yang sistematis mengenai komputasi semantik dan aplikasi AI modern. Memberi materi ajar yang dapat diadaptasi pada tingkat sarjana dan pascasarjana. Memberikan panduan praktis bagi pengembang sistem mengenai desain, evaluasi, dan implementasi solusi berbasis pengetahuan dan pembelajaran mendalam. Menstimulasi dialog interdisipliner antara ilmuwan komputer, insinyur, praktisi bisnis, dan pembuat kebijakan mengenai pengembangan AI yang andal, transparan, dan bertanggung jawab.

Penyusunan buku ini melibatkan kontribusi substantif dari banyak pihak. Kami menyampaikan penghargaan setinggi-tingginya kepada para penulis bab yang telah menyumbangkan tulisan berbobot, hasil eksperimen, dan analisis kritis. Reviewer internal dan eksternal yang memberikan umpan balik konstruktif untuk memperbaiki mutu naskah. Rekan-rekan laboratorium dan institusi yang menyediakan sumber daya komputasi dan dataset eksperimen. Mitra industri dan praktisi yang berbagi pengalaman implementasi nyata dan tantangan lapangan. Keluarga dan kolega yang memberikan dukungan moral selama proses penulisan dan penyuntingan.

Transformasi pengetahuan menuju kecerdasan buatan bukan hanya soal peningkatan kapasitas komputasi atau kompleksitas model. Proses ini memerlukan integrasi konsep, teknik, dan prinsip etis sehingga sistem yang dihasilkan mampu berfungsi secara dapat dipertanggungjawabkan dalam konteks sosial dan operasional. Kami berharap buku ini menjadi kontribusi bermakna dalam upaya tersebut—memberi landasan intelektual, alat metodologis, dan inspirasi praktis bagi pengembangan AI yang lebih cerdas, transparan, dan berguna bagi masyarakat luas.

Semangat & Selamat Membaca.....!!!!!

Semarang, Juni 2026
Penulis

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

DAFTAR ISI

KATA PENGANTAR ..... **i**
DAFTAR ISI ..... **iv**
BAB 1 LLM: MASA LALU, JANJI, DAN MASALAHNYA **1**
1.1 Pendahuluan 1
1.2 Fondasi Teknis Di Balik Jaringan Saraf Modern 10
1.3 Membangun Model Bahasa Yang Besar 20
1.4 Menuju Model Bahasa Besar Berbasis Transformer 25
1.5 Transformer Dalam Praktik 34
1.6 Kesenjangan Antara Inovasi Dan Akuntabilitas Dalam AI 41
BAB 2 MASALAH, SOLUSI, DAN KOMPUTASI SEMANTIK **43**
2.1 Integrasi Masalah Dan Penawaran Melalui Komputasi Semantik 43
2.2 Mekanisme Integrasi Dalam Ekosistem Pertukaran Solusi 44
2.3 Mendeskripsikan Solusi Dalam SCDL 45
2.4 Mendeskripsikan Masalah Dalam SQDL 48
2.5 SNL Sebagai Antarmuka ORA Bagi Pengguna 49
2.6 Proses Mencocokkan Masalah Dan Solusi 49
2.7 Definisi Dan Ruang Lingkup Komputasi Semantik 50
2.8 Evaluasi Komposisi Layanan Dan Prospeknya 51
BAB 3 KLASIFIKASI TEKS BERBASIS PEMBOBOTAN RANDOM WALK **53**
3.1 Optimasi Pembobotan Istilah Melalui Pendekatan Random Walk 53
3.2 Algoritma Random Walk 54
3.3 Random Walk Untuk Pembobotan Istilah 55
3.4 Pengaturan Eksperimental 58
3.5 Evaluasi Model Random Walk Untuk Klasifikasi Teks 60
3.6 Evaluasi Akhir Dan Prospek Pengembangan 67
BAB 4 PENALARAN PEER-TO-PEER UNTUK ONTOLOGI YANG SALING TERHUBUNG **68**
4.1 Penalaran Terdistribusi Untuk Ontologi Web Terhubung 68
4.2 Konteks Dan Referensi Pengembangan 72
4.3 Mendistribusikan Resolusi Logis 74
4.4 Resolusi Terdistribusi Untuk ALC 78
4.5 Evaluasi Metode Resolusi Terdistribusi 84
4.6 Arah Pengembangan Metode Resolusi Terdistribusi 94
4.7 Kesimpulan Dan Arah Pengembangan Penalaran Terdistribusi 96
BAB 5 DETEKSI ADEGAN VIDEO MULTIMODAL **97**
5.1 Deteksi Adegan Video Berbasis Struktur Shot Dan Multimodal 97
5.2 Metode Segmentasi Dan Deteksi Adegan Video 98
5.3 Pengelompokan Sekuensial Optimal 100
5.4 Kompleksitas Komputasi 103

5.5 Deteksi Adegan Video Terbatas 104
5.6 Menggunakan Berbagai Modalitas 106
5.7 Pengelompokan Temporal Adegan Video Berbasis Fitur Arbitrer 110
BAB 6 AI BERBASIS PENGETAHUAN UNTUK ANALISIS PROSES BISNIS 111
6.1 XAI Berbasis Knowledge Graph Untuk Analisis Proses Bisnis 111
6.2 AI Berbasis Pengetahuan 113
6.3 Analisis Proses Berbasis Grafik Pengetahuan 116
6.4 Permodelan Desain Pohon Keputusan Stokastik 119
6.5 Justifikasi Tujuan Desain 123
6.6 Integrasi Knowledge Graph Dan Machine Learning Untuk XAI-PA 132
BAB 7 PEMBELAJARAN MENDALAM 134
7.1 Perkembangan AI: Dari Machine Learning Hingga Deep Learning 134
7.2 Arsitektur Pembelajaran Mendalam 136
7.3 Algoritma Pelatihan 142
7.4 Kerangka Kerja Deep Learning 148
7.5 Pembelajaran Mendalam Dan Big Data 149
7.6 Aplikasi Deep Learning 150
7.7 Hambatan Deep Learning 152
BAB 8 JARINGAN KONVOLUSIONAL UNTUK PENGENALAN KONSEP VISUAL 154
8.1 Perkembangan Dan Keunggulan CNN 154
8.2 Evolusi Metode Pengenalan Visual 155
8.3 Jaringan Neural Konvolusional Dalam 156
8.4 Evaluasi Komparatif Ekstraksi Fitur Dan Fusi Pada Benchmark Gambar 165
8.5 Sistem Pengenalan Konsep Visual Berbasis Fusi Fitur Pada Arsitektur CNN 168
BAB 9 JARINGAN NEURAL DALAM UNTUK SOLUSI INDUSTRI REAL-TIME 170
9.1 Pembelajaran Mendalam Dan NAS 170
9.2 Operasi Graf Di Bawah NEAT 173
9.3 CA Untuk Evolusi Grafik 175
9.4 Mekanisme KD 179
9.5 Masalah Benchmark Game 2D 181
9.6 Masalah Benchmark Pelacakan Gerak Trailer 191
9.7 Kesesuaian CA Untuk Sistem Kompleks 198
BAB 10 PENGENALAN EMOSI AUDIO BERBASIS FUSI MULTIMODAL 200
10.1 Dasar Fusi Informasi Multimodal 200
10.2 Isu-Isu Kritis Fusi Informasi 200
10.3 Reduksi Dimensi Dan Fusi Fitur Berbasis Kernel 202
10.4 Teori Entropi Dan Metode Kernel 203
10.5 Transformasi Fitur Berdasarkan Pemeriksaan Komponen Entropi Kernel 205
10.6 Pengenalan Emosi Audio Berdasarkan Komponen Entropi Kernel 207
10.7 Hasil Dan Evaluasi Eksperimen Sistem Pengenalan Emosi 210
10.8 Kesimpulan Kontribusi KECA 215

BAB 11 DETEKSI PERISTIWA VIDEO BERBASIS ENSEMBLE DEEP LEARNING 216
11.1 Latar Belakang Deteksi Peristiwa Video 216
11.2 Deteksi Peristiwa Video 218
11.3 Kerangka Kerja Ensemble Deep Learning 219
11.4 Evaluasi Kinerja Ensemble Deep Learning 226
11.5 Deteksi Peristiwa Video Berbasis Deep Learning Ensemble 235
BAB 12 EKSTRAKSI PENGETAHUAN BERBASIS DEEP LEARNING PADA DATA MEDIS 237
12.1 Pembelajaran Mendalam Untuk Analisis Data Medis 237
12.2 Metode Pembelajaran Adaptif DBN 239
12.3 Ekstraksi Pengetahuan Untuk DBN Adaptif 242
12.4 Evaluasi Adaptive DBN Pada Data Medis 246
12.5 Ringkasan Dan Evaluasi Metode Adaptive DBN 253
BAB 13 DETEKSI FRAUD BERBASIS GRAF: TREN, TANTANGAN, DAN PROSPEK 254
13.1 Komputasi Graf Untuk Deteksi Kejahatan Keuangan 254
13.2 Tren Dan Karakteristik Penipuan Keuangan 255
13.3 Tantangan Komputasi Graf Pada Deteksi Fraud 264
13.4 Kesimpulan Dan Tantangan Deteksi Fraud Berbasis Graf 270
BAB 14 PEMBUATAN JUDUL OTOMATIS BERBASIS TRANSFORMER PRETRAINED 272
14.1 Pembuatan Judul Otomatis Untuk Sumber Belajar Daring 272
14.2 Metode Pembuatan Judul Otomatis 274
14.3 Pendekatan Generasi Judul Dengan Fine-Tuning GPT-2 276
14.4 Eksperimen Dan Hasil Untuk Pembuatan Judul 283
14.5 Eksperimen Dan Hasil Untuk Modul Peringkas 287
14.6 Pembuatan Judul Hierarkis Untuk Jalur Pembelajaran 290
14.7 Pembuatan Judul Otomatis Dengan Transformer 294
DAFTAR PUSTAKA 295

BAB 1
LLM: MASA LALU, JANJI, DAN MASALAHNYA

1.1 PENDAHULUAN

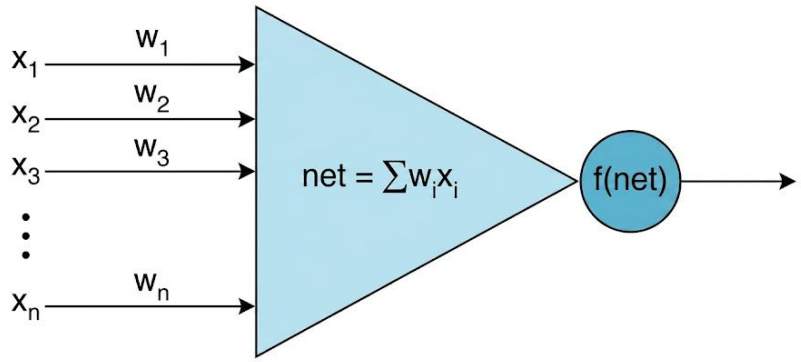
Model bahasa besar berbasis transformer saat ini berada di garis depan kecerdasan buatan modern. Keunggulannya berawal dari artikel penting *Attention is All You Need*. Vaswani dan rekan-rekannya menyarankan untuk menempatkan mekanisme perhatian di dalam modul encoder dan decoder dari autoencoder daripada menggunakannya untuk fokus di antara kedua modul ini. Dalam artikel ini, peprtama-tama kami menyajikan wawasan penting dari AI awal yang mengarah ke pembelajaran mendalam. Kemudian kami menjelaskan alat-alat matematika yang diperlukan untuk memahami generasi LLM saat ini dan diikuti dengan deskripsi singkat tentang arsitektur transformer. Kemudian kami memberikan contoh LLM dalam aksi dan menyimpulkan dengan beberapa pengamatan tentang janji dan masalahnya.

Jaringan saraf, yang sering digambarkan sebagai komputasi yang terinspirasi saraf, atau pemrosesan terdistribusi paralel, mengurangi penekanan pada penggunaan simbol dan penalaran berbasis logika secara eksplisit. Pendekatan jaringan saraf dirancang untuk menangkap hubungan dan asosiasi dalam domain aplikasi dan menginterpretasikan situasi baru dalam konteks pola relasional yang telah dipelajari sebelumnya. Filosofi jaringan saraf menduga bahwa kecerdasan muncul dalam sistem komponen sederhana yang berinteraksi, neuron biologis atau buatan. Hal ini terjadi melalui proses pembelajaran atau adaptasi di mana koneksi antar komponen disesuaikan saat pola di dunia diproses. Komputasi dalam sistem ini didistribusikan di seluruh kumpulan, atau lapisan, neuron. Pemecahan masalah bersifat paralel dalam arti bahwa semua neuron dalam kumpulan atau lapisan memproses input mereka secara simultan dan independen. Sistem ini juga mengalami degradasi secara bertahap karena informasi dan pemrosesan didistribusikan di seluruh node dan lapisan dan tidak terlokalisasi pada satu komponen jaringan.

Algoritma dan arsitektur yang mengimplementasikan teknik koneksionis biasanya dilatih atau dikondisikan daripada diprogram secara eksplisit. Fakta ini merupakan kekuatan utama dari pendekatan ini, karena arsitektur jaringan dan algoritma pembelajaran yang dirancang dengan tepat seringkali dapat menangkap invariansi di dunia, bahkan dalam bentuk attractor aneh, tanpa diprogram secara eksplisit untuk mengenalinya.

Dasar dari sebuah jaringan adalah neuron buatan, seperti yang ditunjukkan pada Gambar 1.1 Komponen minimal dari neuron buatan adalah:

1) *Sinyal input* X_i. Sinyal-sinyal ini dapat berasal dari lingkungan atau dari aktivasi neuron lain. Model yang berbeda bervariasi dalam rentang nilai input yang diperbolehkan; biasanya, input bersifat diskrit, dari himpunan $\{0, 1\}$ atau $\{-1, 1\}$.

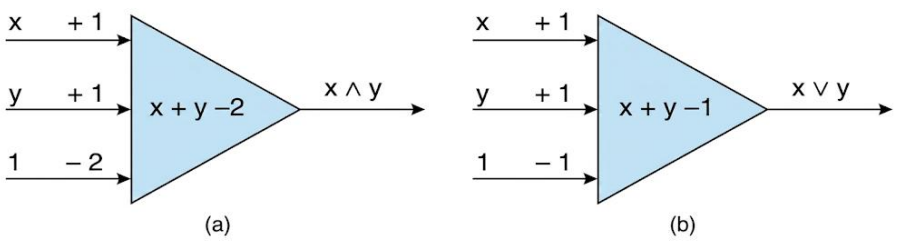


Gambar 1.1 Neuron buatan dengan vektor input x_i, bobot w_i untuk setiap input, dan fungsi ambang batas f yang menentukan nilai output neuron.

2) Seperangkat bobot bernilai $riil, w_i$. Nilai-nilai ini menggambarkan kekuatan koneksi.
3) *Tingkat aktivasi*, $\sum\ w_i x_i$. Tingkat aktivasi neuron ditentukan oleh kekuatan kumulatif sinyal inputnya di mana setiap sinyal diskalakan, atau dikalikan, dengan bobot koneksi yang terkait dengan input tersebut. Tingkat aktivasi dihitung dengan mengambil jumlah input berbobot, yaitu $\sum w_i x_i$. Sigma Yunani, C, menunjukkan bahwa nilai-nilai ini dijumlahkan.
4) *Ambang batas atau fungsi pemetaan nonlinier terbatas*, f. Fungsi ambang batas menghitung output neuron dengan melihat apakah nilainya di atas tingkat aktivasi. Fungsi pemetaan nonlinier menghasilkan respons hidup/mati atau respons bertingkat untuk neuron tersebut.

Contoh awal komputasi neural adalah neuron McCulloch–Pitts. Input dari neuron-neuron ini adalah +1, yaitu $benar$, atau —1, $salah$. Fungsi aktivasi mengalikan setiap input dengan bobot yang sesuai dan menambahkan hasilnya; jika jumlah ini lebih besar dari atau sama dengan nol, neuron mengembalikan 1, benar, jika tidak, —1, salah. McCulloch dan Pitts menunjukkan bagaimana neuron-neuron ini dapat dibangun untuk menghitung fungsi logika apa pun.

Gambar 1.2 menunjukkan neuron McCulloch–Pitts untuk menghitung fungsi logika $dan\ (\Lambda)$ dan $atau\ (V)$. Neuron dan, (a) di sebelah kiri, memiliki tiga input: x dan y adalah nilai yang akan digabungkan; input ketiga, terkadang disebut $bias$, memiliki nilai konstan +1. Data input dan bias memiliki bobot masing-masing +1, +1, dan —2. Dengan demikian, untuk setiap nilai input x dan y, neuron menghitung $x\ +\ y\ -\ 2$. Tabel 1.1 menunjukkan bahwa jika nilai ini kurang dari 0, ia mengembalikan —1, salah, jika tidak, 1, benar.



Gambar 1.2 Neuron McCulloch–Pitts untuk dan (a), dan atau (b).

Tabel 1.1 Model McCulloch–Pitts untuk menghitung logika dan dari Gambar 2(a).

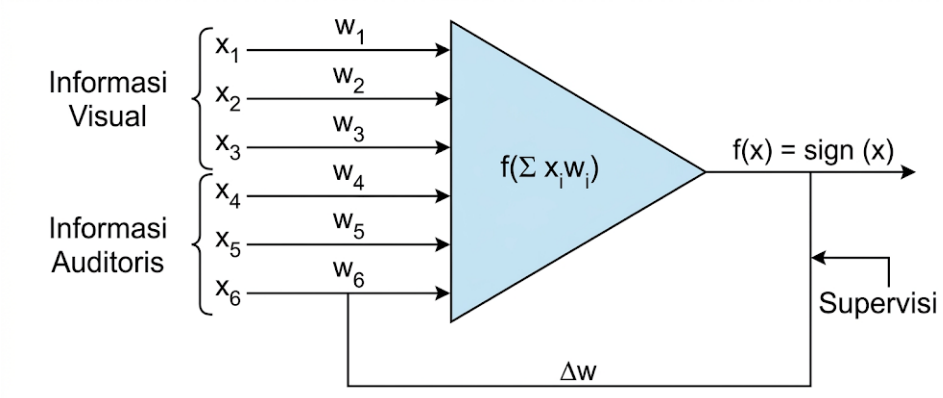
x	y	$x = y - 2$	**Output**
1	1	0	1
1	0	-1	-1
0	1	-1	-1
0	0	-2	-1

Neuron atau (b) di sebelah kanan, menggambarkan neuron yang menghitung $x\ V\ y$. Jumlah tertimbang data masukan untuk neuron V lebih besar dari atau sama dengan 0 kecuali jika x dan y sama dengan -1, yaitu salah. Meskipun McCulloch dan Pitts menunjukkan kekuatan komputasi saraf, minat dalam pembelajaran jaringan saraf baru mulai berkembang dengan pengembangan algoritma pembelajaran praktis. Model pembelajaran awal sangat bergantung pada karya psikolog Donald Hebb, yang berspekulasi bahwa pembelajaran terjadi di otak melalui modifikasi, atau pengkondisian, sinapsis. Hebb menyatakan:

> Ketika akson sel A cukup dekat untuk merangsang sel B dan berulang kali atau terus-menerus memicunya, beberapa proses pertumbuhan atau perubahan metabolisme terjadi pada satu atau kedua sel sehingga efisiensi A, sebagai salah satu sel yang memicu B, meningkat.

Pembelajaran fisiologi saraf telah mengkonfirmasi gagasan Hebb bahwa kedekatan temporal dari pelepasan impuls neuron yang terhubung dapat memodifikasi kekuatan sinaptik mereka, meskipun dengan cara yang lebih kompleks daripada intuisi Hebb tentang "peningkatan efisiensi". Selanjutnya, kita akan mendemonstrasikan pembelajaran *Hebbian*, yang termasuk dalam kelas hukum pembelajaran koinsiden. Pembelajaran ini menghasilkan perubahan bobot sebagai respons terhadap peristiwa lokal dalam pemrosesan saraf.

Pembelajaran jaringan saraf dapat berupa tanpa pengawasan, dengan pengawasan, atau kombinasi hibrida dari keduanya. Contoh yang telah kita lihat sejauh ini adalah tanpa pengawasan, karena jaringan dan bobotnya mengubah sinyal input menjadi nilai output yang diinginkan. Sekarang kita akan mempertimbangkan contoh pembelajaran Hebbian tanpa pengawasan di mana setiap output memiliki faktor penyesuaian bobot. Dalam pembelajaran tanpa pengawasan, kritikus tidak tersedia untuk memberikan nilai output yang "benar". Akibatnya, bobot harus dimodifikasi di beberapa iterasi semata-mata sebagai fungsi dari nilai input dan output neuron. Pelatihan jaringan Hebbian pada Gambar 1.3 memiliki efek memperkuat respons jaringan terhadap pola yang telah dilihatnya dan menafsirkan pola baru dengan tepat.



Gambar 1.3 Jaringan Hebbian, tanpa pengawasan ekstra-jaringan, yang mempelajari respons untuk stimulus tak terkondisi. ΔW menyesuaikan bobot pada setiap iterasi data melalui jaringan. Fungsi ambang ditunjukkan pada Gambar 1.7(a).

Gambar 1.3 menunjukkan bagaimana teknik Hebbian dapat digunakan untuk memodelkan pembelajaran respons terkondisi, di mana stimulus yang dipilih secara acak dapat digunakan untuk mengkondisikan respons yang diinginkan. Eksperimen klasik Pavlov pada tahun 1890-an menawarkan contoh respons terkondisi. Seekor anjing diberi makan sementara bel dibunyikan. Anjing itu mengeluarkan air liur sebagai antisipasi makanannya. Respons tak terkondisi dari hewan yang mengeluarkan air liur adalah kehadiran makanan. Setelah beberapa kali kedatangan makanan disertai dengan bunyi bel, bel dibunyikan tanpa makanan. Anjing itu mengeluarkan air liur. Bunyi bel menghasilkan respons terkondisi anjing!

Gambar 1.3 menunjukkan bagaimana jaringan Hebbian dapat mentransfer respons dari stimulus primer atau tak terkondisi ke stimulus terkondisi. Dalam eksperimen Pavlov, respons air liur anjing terhadap makanan ditransfer ke bel. Penyesuaian bobot, ΔW, pada setiap iterasi jaringan, dijelaskan oleh persamaan:

$$\Delta W = c * f(X, W) * X.$$

Dalam persamaan ini, c adalah konstanta pembelajaran, sebuah bilangan desimal positif kecil, yang penggunaannya memodulasi tingkat pembelajaran pada setiap langkah, seperti yang dijelaskan lebih detail pada Gambar 1.3, $f(X, W)$ adalah output jaringan pada setiap iterasi, dan X_i adalah vektor input pada iterasi tersebut.

Jaringan pada Gambar 1.3 memiliki dua lapisan, lapisan input dengan enam node dan lapisan output dengan satu node. Lapisan output mengembalikan +1, yang menandakan bahwa neuron output telah aktif, atau -1 yang menandakan bahwa neuron tersebut belum aktif. Umpan balik, Supervise, yang memantau jaringan, ΔW, mengambil setiap output jaringan dan mengalikannya dengan vektor input dan konstanta pembelajaran untuk menghasilkan kumpulan bobot untuk vektor input pada iterasi jaringan berikutnya. Kami menetapkan konstanta pembelajaran ke bilangan real positif kecil, 0,2. Dalam contoh ini, kita melatih jaringan pada pola $[1, -1, 1, -1, 1\ -1]$ yang menggabungkan dua pola, $[1, -1, 1]$ dan

$[-1, 1, -1]$. Pola $[1, -1, 1]$ mewakili stimulus tak terkondisi dan $[-1, 1, -1]$ mewakili stimulus baru.

Asumsikan bahwa jaringan sudah merespons secara positif terhadap stimulus tak terkondisi tetapi netral terhadap stimulus baru. Kita mensimulasikan respons positif jaringan terhadap stimulus tak terkondisi dengan vektor bobot $[1, -1, 1]$ yang persis sesuai dengan pola input. Respons netral jaringan terhadap stimulus baru disimulasikan oleh vektor bobot $[0, 0, 0]$. Menggabungkan kedua vektor ini memberikan vektor bobot awal untuk jaringan, $[1, -1, 1, 0, 0, 0]$. Jaringan kemudian dilatih pada pola input, dengan harapan dapat menghasilkan konfigurasi bobot yang akan menghasilkan respons jaringan positif terhadap stimulus baru. Iterasi pertama dari jaringan tersebut menghasilkan hasil sebagai berikut:

$$\begin{aligned} \mathrm{W^*X\ \&} &= (1^*1) + (-1^* - 1) + (1^*1) + (0^* - 1) + (0^*1) + (0^* - 1) \\ &= (1) + (1) + (1) = 3, \text{and } \mathrm{f}(3) = \text{sign}\,(3) = 1. \end{aligned}$$

Sekarang jaringan Hebbian membuat vektor bobot baru W^2:

$$\begin{aligned} W^2 &= [1, -1, 1, 0, 0, 0] + 0.2^*(1)^*[1, -1, 1, -1, 1, -1] \\ &= [1, -1, 1, 0, 0, 0] + [0.2, -0.2, 0.2, -0.2, 0.2, -0.2] \\ &= [1.2, -1.2, 1.2, -0.2, 0.2, -0.2]. \end{aligned}$$

Selanjutnya, jaringan yang telah disesuaikan melihat pola input asli dengan bobot baru:

$$\begin{aligned} W^*X &= (1.2^*1) + (-1.2^* - 1) + (1.2^*1) + (-0.2^* - 1) + (0.2^*1) + (-0.2^* - 1) \\ &= (1.2) + (1.2) + (1.2) + (0.2) + (0.2) + (0.2) \\ &= 4.2, \text{and sign}\,(4.2) = 1. \end{aligned}$$

Sekarang jaringan Hebbian membuat vektor bobot baru W^3:

$$\begin{aligned} W^3 &= [1.2, -1.2, 1.2, -0.2, 0.2, -0.2] + 0.2^*(1)^*[1, -1, 1, -1, 1, -1] \\ &= [1.2, -1.2, 1.2, -0.2, 0.2, -0.2] + [0.2, -0.2, 0.2, -0.2, 0.2, -0.2] \\ &= [1.4, -1.4, 1.4, -0.4, 0.4, -0.4]. \end{aligned}$$

Sekarang dapat dilihat bahwa hasil perkalian vektor bobot, $W * X$, akan terus tumbuh ke arah positif, dengan nilai setiap elemen vektor bobot meningkat sebesar 0.2 ke arah + atau – pada setiap siklus pelatihan. Setelah 10 iterasi pelatihan Hebbian lagi, vektor bobot akan menjadi...

$$W^{13} = [3.4, -3.4, 3.4, -2.4, 2.4, -2.4].$$

Kami menggunakan vektor bobot terlatih ini untuk menguji respons jaringan terhadap dua pola parsial. Kami ingin melihat apakah jaringan terus merespons stimulus tak terkondisi secara positif dan, yang lebih penting, apakah jaringan sekarang telah memperoleh respons positif terhadap stimulus terkondisi yang baru. Kami menguji jaringan terlebih dahulu pada

stimulus tak terkondisi $[1, -1, 1]$. Kami mengisi tiga argumen terakhir dari vektor input dengan penugasan acak 1 dan —1, misalnya, kami menguji jaringan pada vektor $[1, -1, 1, 1, 1, -1]$:

$$\begin{aligned}\text{sign}(\text{W}^*\text{X}) &= \text{sign}((3.4^*1) + (-3.4^* - 1) + (3.4^*1) + (-2.4^*1) \\ &\quad + (2.4^*1) + (-2.4^* - 1)) \\ &= \text{sign}(3.4 + 3.4 + 3.4 - 2.4 + 2.4 + 2.4) \\ &= \text{sign}(12.6) = +1.\end{aligned}$$

Jaringan tersebut masih merespons secara positif terhadap stimulus tak terkondisi asli. Selanjutnya, kita melakukan tes kedua menggunakan stimulus tak terkondisi asli dan vektor acak yang berbeda pada tiga posisi terakhir $[1, -1, 1, 1, -1, -1]$:

$$\begin{aligned}\mathbf{s}\text{ign}(W^*X) &= \text{sign}((3.4^*1) + (-3.4^* - 1) + (3.4^*1) + (-2.4^*1) \\ &\quad + (2.4^* - 1) + (-2.4^* - 1)) \\ &= \text{sign}(3.4 + 3.4 + 3.4 - 2.4 - 2.4 + 2.4) \\ &= \text{sign}(7.8) = +1.\end{aligned}$$

Vektor kedua juga menghasilkan respons jaringan yang positif. Dengan dua contoh ini, sensitivitas jaringan terhadap stimulus asli telah diperkuat, karena paparan berulang terhadap stimulus tersebut. Selanjutnya, kita menguji respons jaringan terhadap pola stimulus baru, $[-1, 1, -1]$, yang dikodekan dalam tiga posisi terakhir vektor input. Kita mengisi tiga posisi vektor pertama dengan penugasan acak dari himpunan $\{1, -1\}$ dan menguji jaringan pada vektor $[1, 1, 1, -1, 1, -1]$:

$$\begin{aligned}\text{sign}(W^*X)\ \& &= \text{sign}((3.4^*1) + (-3.4^* - 1) + (3.4^*1) + (-2.4^*1) \\ &\quad + (2.4^*1) + (-2.4^* - 1)) \\ &= \text{sign}(3.4 - 3.4 + 3.4 + 2.4 + 2.4 + 2.4) \\ &= \text{sign}(10.6) = +1.\end{aligned}$$

Kami melakukan satu eksperimen terakhir, dengan pola vektor yang sedikit terdegradasi. Ini dapat mewakili situasi stimulus di mana sinyal input diubah, mungkin karena makanan baru dan/atau bel yang berbunyi berbeda digunakan. Kami menguji jaringan pada vektor input $[1, -1, -1, 1, 1, -1]$, di mana tiga parameter pertama berbeda satu digit dari stimulus tak terkondisi asli dan tiga parameter terakhir berbeda satu digit dari stimulus terkondisi:

$$\begin{aligned}\text{sign}(\text{W}^*\text{X})\ \& &= \text{sign}((3.4^*1) + (-3.4\ \text{\^{}} * -1) + (3.4^*1) + (-2.4^*1) \\ &\quad + (2.4^*1) + (-2.4^* - 1)) \\ &= \text{sign}(3.4 - 3.4 + 3.4 + 2.4 + 2.4 + 2.4) \\ &= \text{sign}(10.6) = +1.\end{aligned}$$

Bahkan stimulus yang sebagian terdegradasi ini pun dikenali. Apa yang dihasilkan oleh model pembelajaran Hebbian? Kami menciptakan asosiasi antara stimulus baru dan respons lama

dengan berulang kali menyajikan stimulus lama dan baru secara bersamaan. Jaringan belajar untuk mentransfer responsnya ke stimulus baru tanpa pengawasan eksternal apa pun. Sensitivitas yang diperkuat ini juga memungkinkan jaringan untuk merespons dengan cara yang sama terhadap versi stimulus yang sedikit terdegradasi. Ini dicapai dengan menggunakan pembelajaran koinsidensi Hebbian untuk meningkatkan kekuatan respons jaringan terhadap pola keseluruhan. Ini meningkatkan kekuatan pada setiap komponen individual dari pola tersebut: contoh pengorganisasian diri yang muncul dari penggunaan aturan Hebb. Pola stimulus sekunder juga dikenali!

$$= \text{sign}(10.6) = +1.$$

Pada tahun 1958 Rosenblatt, menciptakan Perceptron, sebuah perangkat elektronik yang terinspirasi oleh prinsip-prinsip neurologis. Rosenblatt adalah seorang psikolog dan ahli saraf yang, pada tahun 1959, menjadi direktur Program Pembelajaran Sistem Kognitif Cornell. Jaringan perceptron terdiri dari satu lapisan neuron perceptron N yang diaktifkan oleh n input masing-masing dengan bobot, w_n, seperti yang ditunjukkan pada Gambar 1.3 Artikel Rosenblatt tahun 1962 yang menjelaskan perceptron berjudul Prinsip-prinsip neurodinamika: Perceptron dan teori mekanisme otak.

Menariknya, banyak peneliti awal, para pendahulu teknologi jaringan saraf, mengklaim inspirasi dari aktivitas saraf manusia untuk kreasi mereka. Pada abad ke-19, Boole, pencipta sistem aljabar yang mendukung komputasi modern, dan yang logikanya pertama kali diotomatisasi oleh neuron McCulloch–Pitts, menawarkan contoh utama. Dalam bab pertama dari Investigasi Hukum Pikiran, yang menjadi dasar Teori Matematika Logika dan Probabilitas, Boole menggambarkan tujuannya sebagai

> ... untuk menyelidiki hukum-hukum fundamental dari operasi pikiran yang digunakan untuk melakukan penalaran: untuk mengungkapkannya dalam bahasa simbolis Kalkulus, dan atas dasar ini untuk membangun ilmu logika dan menginstruksikan metodenya; ... dan akhirnya untuk mengumpulkan dari berbagai elemen kebenaran yang terungkap dalam penyelidikan ini beberapa petunjuk yang mungkin mengenai sifat dan konstitusi pikiran manusia.

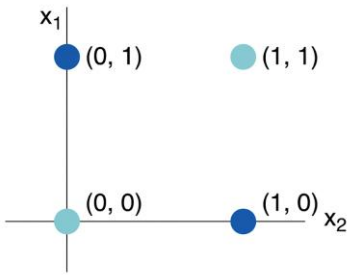
Perceptron awalnya disambut dengan antusias. Namun, Nilsson dan lainnya menganalisis keterbatasan model perceptron. Mereka menunjukkan bahwa perceptron tidak dapat menyelesaikan kelas masalah yang sulit, yaitu masalah di mana titik data tidak dapat dipisahkan secara linier dalam dimensi pernyataan masalah asli. Meskipun berbagai peningkatan model perceptron, termasuk perceptron berlapis, telah dipertimbangkan pada saat itu, Minsky dan Papert, dalam buku mereka Perceptrons, berpendapat bahwa masalah keterpisahan linier tidak dapat diatasi oleh bentuk jaringan perceptron yang ada saat itu. Contoh klasifikasi yang tidak dapat dipisahkan secara linier adalah masalah eksklusif-atau pada Tabel 1.2

Pertimbangkan perceptron dengan dua input, x_1, x_2, dua bobot, w_1, w_2, dan ambang batas t. Untuk mempelajari fungsi ini, sebuah jaringan harus menemukan penugasan bobot yang memenuhi ketidaksetaraan berikut, yang terlihat secara grafis pada Gambar 1.4:

$w1 * 1 + w2 * 1 < t$, dari baris 1 tabel kebenaran,
$w1 * 1 + 0 > t$, dari baris 2 tabel kebenaran,
$0 + w2 * 1 > t$, dari baris 3 tabel kebenaran,
$0 + 0 < t$, atau t harus positif, dari baris terakhir tabel.

Tabel 1.2 Tabel kebenaran untuk operator eksklusif-atau.

x_1	x_2	Output
1	1	0
1	0	1
0	1	1
0	0	0



Gambar 1.4 Masalah eksklusif-atau. Tidak ada garis lurus pada grid dua dimensi yang dapat memisahkan titik data (0, 1) dan (1, 0) dari (0, 0) dan (1, 1).

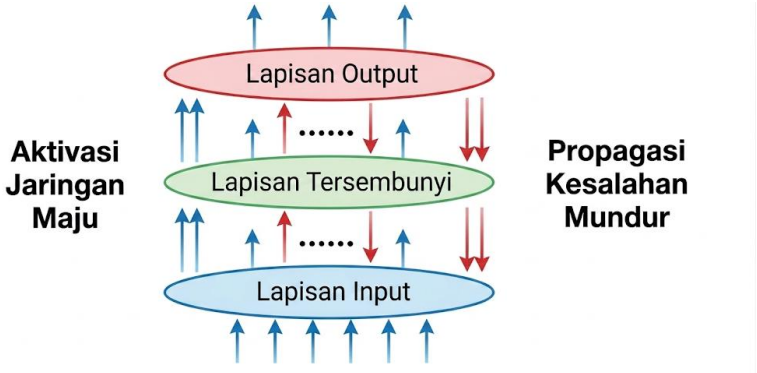
Rangkaian persamaan pada w_1, w_2 dan t ini tidak memiliki solusi, membuktikan bahwa perceptron yang dapat menyelesaikan masalah eksklusif-atau adalah tidak mungkin. Meskipun jaringan multilayer pada akhirnya akan dibangun yang dapat menyelesaikan masalah eksklusif, algoritma pembelajaran perceptron hanya bekerja untuk jaringan lapisan tunggal. Yang membuat eksklusif-atau tidak mungkin bagi perceptron adalah bahwa kedua kelas yang akan dibedakan tidak dapat dipisahkan secara linier. Hal ini dapat dilihat pada Gambar 1.4 Tidak mungkin untuk menggambar garis lurus dalam dua dimensi yang memisahkan titik data $\{(0,0),(1,1)\}$ dari $\{(0,1),(1,0)\}$.

Kita dapat menganggap himpunan nilai data untuk suatu jaringan sebagai mendefinisikan suatu ruang. Setiap parameter data masukan sesuai dengan satu dimensi, dengan setiap nilai masukan mendefinisikan sebuah titik dalam ruang. Dalam contoh exclusive-or, keempat nilai masukan, yang diindeks oleh koordinat x_1, x_2, membentuk titik-titik data pada Gambar 1.4 Masalah mempelajari klasifikasi biner dari instance pelatihan direduksi menjadi memisahkan titik-titik ini menjadi dua kelompok. Untuk ruang n dimensi, klasifikasi dapat dipisahkan secara linier jika kelas-kelasnya dapat dipisahkan oleh hyperplane $(n-1)$

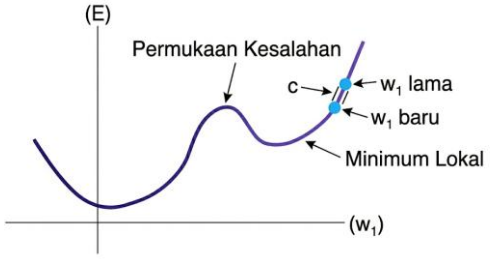
dimensi. Dalam dua dimensi, hyperplane $(n-1)$ dimensi adalah garis; dalam tiga dimensi, itu adalah bidang, dan seterusnya.

Sebagai hasil dari keterbatasan pemisahan linier, pembelajaran bergeser ke arah pekerjaan dalam arsitektur berbasis simbol, memperlambat kemajuan dalam metodologi koneksionis. Namun, pekerjaan selanjutnya pada tahun 1980-an dan 1990-an telah menunjukkan bahwa masalah-masalah ini dapat dipecahkan; Selanjutnya kita akan membahas backpropagation, sebuah perluasan pembelajaran perceptron yang bekerja untuk jaringan berlapis-lapis.

Neuron-neuron dalam jaringan ini, seperti yang terlihat pada perceptron multi-lapisan pada Gambar 1.5, terhubung dalam lapisan-lapisan, dengan unit-unit dalam lapisan n hanya meneruskan aktivasinya ke neuron-neuron di lapisan $n+1$. Pemrosesan sinyal multi-lapisan berarti bahwa kesalahan yang terjadi jauh di dalam jaringan dapat menyebar dan berkembang dengan cara yang kompleks dan tidak terduga di seluruh lapisan. Dengan demikian, analisis sumber kesalahan keluaran akhir dan menghubungkan kesalahan tersebut kembali ke node jaringan yang menghasilkannya menjadi kompleks. Backpropagation adalah algoritma untuk membagi tanggung jawab ini dan menyesuaikan bobot jaringan sesuai dengan itu.



Gambar 1.5 Skema untuk jaringan saraf multi-lapisan tersembunyi. Propagasi kesalahan mundur ditangani dengan algoritma Bagian 2.1.



Gambar 1.6 Permukaan kesalahan dalam dua dimensi. Dimensi ruang kesalahan suatu masalah adalah jumlah bobot yang terlibat ditambah ukuran kesalahan. Konstanta pembelajaran c mengontrol ukuran langkah yang diambil pada permukaan kesalahan dengan setiap iterasi pembelajaran jaringan. Tujuannya adalah untuk menemukan nilai W_1 di mana E berada pada minimum.

Kemunculan historis jaringan dengan fungsi aktivasi kontinu menunjukkan pendekatan baru untuk pembelajaran pengurangan kesalahan. Misalnya, aturan pembelajaran Widrow–Hoff tidak bergantung pada fungsi aktivasi, meminimalkan kesalahan kuadrat antara nilai keluaran yang diinginkan dan aktivasi jaringan, neti $= \mathrm{WX}_i$. Suatu bentuk Widrow–Hoff menjadi aturan pembelajaran terpenting untuk fungsi aktivasi kontinu. Ini adalah aturan delta.

Secara intuitif, aturan delta didasarkan pada gagasan permukaan kesalahan, seperti yang diilustrasikan pada Gambar 1.6 Permukaan kesalahan ini mewakili kesalahan kumulatif pada suatu kumpulan data sebagai fungsi dari bobot jaringan. Misalnya, jika ada enam bobot yang akan dikondisikan, seperti pada contoh XOR kita di Bagian. 2.1, ukuran kesalahan akan membentuk dimensi ke-7 dari ruang kesalahan. Setiap konfigurasi bobot jaringan diwakili oleh sebuah titik pada permukaan kesalahan n-dimensi ini. Dengan konfigurasi bobot tertentu, kita ingin algoritma pembelajaran kita menemukan arah pada permukaan ini yang paling cepat mengurangi kesalahan. Pendekatan ini disebut pembelajaran penurunan gradien karena gradien adalah ukuran kemiringan, sebagai fungsi arah, dari sebuah titik pada permukaan.

Backpropagation membutuhkan data yang diawasi, misalnya, pengklasifikasi dilatih pada data berlabel. Misalnya, seorang ahli radiologi mungkin memiliki ribuan sinar-X yang mencerminkan tumor sementara sinar-X lainnya bebas tumor. Demikian pula, tukang las mungkin memiliki ribuan contoh las yang dapat diterima dan tidak dapat diterima. Namun, setelah jaringan ini dilatih, mereka akan memindai situasi yang sama sekali baru, contoh yang belum pernah mereka pertimbangkan sebelumnya. Pengklasifikasi harus memutuskan setiap situasi baru dan memberi label sebagai baik atau tidak.

1.2 FONDASI TEKNIS DI BALIK JARINGAN SARAF MODERN

Selanjutnya, kami akan menjelaskan beberapa keterampilan yang diperlukan untuk menangani jaringan saraf dan pembelajaran mendalam secara koheren. Pertama, kami mendemonstrasikan algoritma backpropagation dan menunjukkan solusi untuk masalah XOR yang membatasi penggunaan perceptron generasi pertama. Kedua, kami mendemonstrasikan bagaimana aljabar matriks menyediakan media untuk komputasi jaringan. Terakhir, kami mencatat pentingnya statistik dan secara singkat menjelaskan transformasi softmax. Pembelajaran mendalam, menurut dugaan kami, paling baik digambarkan sebagai statistik komputasi.

Backpropagation: Persamaan diferensial parsial

Pendekatan yang diambil oleh algoritma *backpropagation* adalah memulai dari lapisan keluaran dan menyebarkan kesalahan mundur melalui semua lapisan tersembunyi. Kita tahu bahwa semua informasi yang dibutuhkan untuk memperbarui bobot pada neuron bersifat lokal untuk neuron tersebut, kecuali jumlah kesalahan. Untuk node keluaran, kesalahan ini mudah dihitung sebagai perbedaan antara nilai keluaran yang diinginkan dan nilai keluaran aktual. Untuk node di lapisan tersembunyi, jauh lebih sulit untuk menentukan kesalahan yang menjadi tanggung jawab node tersebut. Fungsi aktivasi untuk back-propagation adalah fungsi logistik (sigmoid):

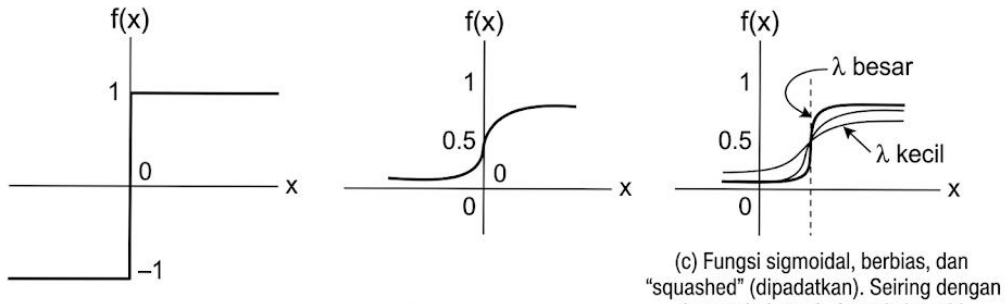
$$f(net) = 1/\left(1 + e^{\{-\lambda * \{net\}\}}\right), \qquad where\ net = \Sigma x_i w_i.$$

Fungsi ini, seperti yang terlihat pada Gambar 1.7(b) dan 1.7(c), digunakan karena empat alasan: Pertama, fungsi ini memiliki bentuk sigmoid yang memberikan keluaran bernilai riil. Kedua, sebagai fungsi kontinu, fungsi ini memiliki turunan di mana-mana. Ketiga, karena nilai turunannya paling besar di tempat fungsi sigmoid berubah paling cepat, maka kesalahan terbesar dikaitkan dengan node yang aktivasinya paling tidak pasti. Terakhir, turunan dari fungsi logistik mudah dihitung dengan pengurangan dan perkalian:

$$f'(net) = \left(1/\left(1 + e^{\{-\lambda * net\}}\right)\right) = \lambda\left(f(net)^*\left(1 - f(net)\right)\right).$$

Pelatihan backpropagation menggunakan aturan delta umum. Untuk node di lapisan tersembunyi, kita melihat kontribusinya terhadap kesalahan di lapisan keluaran. Rumus untuk menghitung penyesuaian bobot w_{ki} pada jalur dari node ke-k ke node ke-i dalam pelatihan back-propagation adalah:

1) $\Delta w_{ki} = -c(d_i - O_i) * O_i(1 - O_i)x_k$, untuk node pada lapisan keluaran, dan
2) $\Delta w_{ki} = -c * O_i(1 - O_i)C_j\left(-delta_j * w_{ij}\right)x_k$, untuk node pada lapisan tersembunyi.



(a) Sebuah fungsi ambang batas linier yang keras dan bipolar.

(b) Sebuah ambang batas sigmoidal dan unipolar.

(c) Fungsi sigmoidal, berbias, dan "squashed" (dipadatkan). Seiring dengan bertambahnya λ, fungsi sigmoid mendekati ambang batas yang keras (*step function*) yang tergeser.

Gambar 1.7 Fungsi ambang batas. (a) digunakan dalam contoh Hebbian. Fungsi sigmoid, kontinu dan terdiferensiasi, digunakan dengan backpropagation.

Dalam (2), j adalah indeks node di lapisan berikutnya tempat sinyal i menyebar dan

$$\partial \mathrm{delta_j} = -\partial \mathrm{Error}/\ \partial \mathrm{net_j} = (\mathrm{d} - \mathrm{O_i}) * \mathrm{O_i}(1 - \mathrm{O_i}).$$

Selanjutnya, kami akan menunjukkan penurunan rumus-rumus ini. Pertama, kami menurunkan (1), rumus untuk penyesuaian bobot pada node di lapisan keluaran. Seperti sebelumnya, yang kita inginkan adalah laju perubahan kesalahan jaringan sebagai fungsi perubahan bobot ke-k, w_k, dari nodei. Kami menunjukkan bahwa

$$\partial Error/\partial w_k = -((d_i - O_i)^* f'(net_i)^* x_k).$$

Karena f, yang bisa berupa fungsi apa saja, sekarang adalah fungsi aktivasi logistik, maka kita memiliki

$$f'(net) = f'(1(1 + e^{-\lambda * net})) = f(net)^*(1 - f(net)).$$

Ingat bahwa $f(\text{net}_i)$ hanyalah O_i. Dengan mensubstitusikan ke persamaan sebelumnya, kita mendapatkan

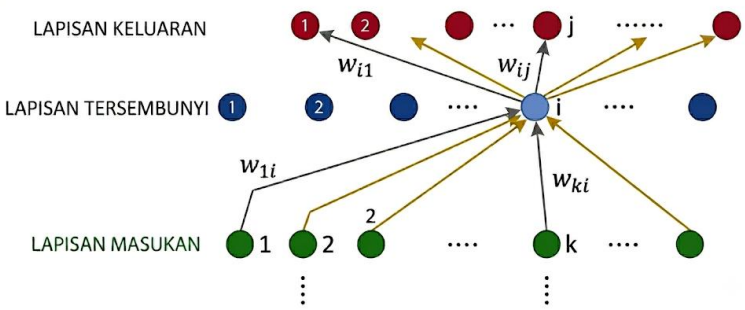
$$\partial Error/\partial w_k = -(d_i - O_i)^* O_i^*(1 - O_i)^* x_k.$$

Karena minimisasi kesalahan mengharuskan perubahan bobot searah dengan komponen gradien negatif, kita kalikan dengan $-c$ untuk mendapatkan penyesuaian bobot untuk node ke-i dari lapisan keluaran:

$$w_k = c(d_i - O_i) * O_i * (1 - O_i) * x_k.$$

Selanjutnya, kita menurunkan penyesuaian bobot untuk node tersembunyi. Demi kejelasan, awalnya kita mengasumsikan satu lapisan tersembunyi. Kita mengambil satu node i pada lapisan tersembunyi dan menganalisis kontribusinya terhadap total kesalahan jaringan. Kita melakukan ini dengan awalnya mempertimbangkan kontribusi node i terhadap kesalahan pada node j di lapisan keluaran. Kemudian kita menjumlahkan kontribusi ini di semua node pada lapisan keluaran. Akhirnya, kita menjelaskan kontribusi bobot input ke-k pada node i terhadap kesalahan jaringan. Gambar 1.8 mengilustrasikan situasi ini. Pertama-tama kita melihat turunan parsial dari kesalahan jaringan terhadap keluaran node i pada lapisan tersembunyi. Kita mendapatkan ini dengan menerapkan aturan rantai:

$$\partial \text{Error}/\partial O_i = \partial \text{Error}/\partial net_j * \partial net_j/\partial O_i.$$



Gambar 1.8 $j - \sum_j -$ delta * w_{ij} adalah kontribusi total node i terhadap kesalahan pada output ke-j. Derivasi kami memberikan penyesuaian untuk w_{ki}.

Negatif dari suku pertama di sisi kanan, $(\delta Error)/(\delta netj)$, disebut $delta_j$. Oleh karena itu, kita dapat menulis ulang persamaan sebagai

$$\partial Error/\partial O_i = -delta_j * \partial net_j \partial O_i.$$

Ingat bahwa aktivasi node j, net_j, pada lapisan keluaran diberikan oleh jumlah hasil perkalian bobotnya sendiri dan nilai keluaran yang berasal dari node pada lapisan tersembunyi:

$$net_j = \sum_i w_{ij} O_i.$$

Karena kita mengambil turunan parsial hanya terhadap satu komponen dari penjumlahan, yaitu koneksi antara node i dan node j, kita peroleh

$$\partial net_j \partial O_i = w_{ij},$$

di mana wi_j adalah bobot pada koneksi dari node i di lapisan tersembunyi ke node j di lapisan keluaran. Dengan mensubstitusikan hasil ini,

$$\partial Error/\partial O_i = -delta_j * w_{ij}.$$

Selanjutnya kita menjumlahkan semua koneksi node i ke lapisan keluaran:

$$\partial Error/\partial O_i = \sum_j -delta_j * w_{ij}.$$

Ini mewakili sensitivitas kesalahan jaringan terhadap keluaran node i pada lapisan tersembunyi. Selanjutnya kita menentukan nilai deltai, yaitu sensitivitas kesalahan jaringan terhadap aktivasi bersih pada node tersembunyi i. Ini memberikan sensitivitas kesalahan jaringan terhadap bobot masukan node i. Dengan menggunakan aturan rantai lagi,

$$-delta_i = \partial Error/\partial net_i = (\partial Error \partial O_i)^* (\partial O_i/\partial net_i).$$

Karena kita menggunakan fungsi aktivasi logistik,

$$\partial O_i \partial net_i = O_i^* (1 - O_i).$$

Sekarang kita substitusikan nilai ini ke dalam persamaan untuk deltai untuk mendapatkan

$$-delta_i = O_i * (1 - O_i) * \sum_j -delta_j * w_{ij}.$$

Terakhir, kita dapat mengevaluasi sensitivitas kesalahan jaringan pada lapisan keluaran terhadap bobot yang masuk pada node tersembunyi i. Kita periksa bobot ke-k pada node i, w_k. Dengan aturan rantai,

$$\partial Error/\partial w_{ki} = (\partial Error \partial net_i * \partial net_i \partial w_{ki}) \\ = -delta_i * (\partial net_i/\partial w_{ki}) = -delta_i * x_k,$$

di mana x_k adalah input ke-k ke node i.
Kita substitusikan nilai $-delta_i$ ke dalam persamaan:

$$\partial Error/\partial w_{ki} = O_i(1 - O_i) * \sum_j (-delta_j * w_{ij}) x_k.$$

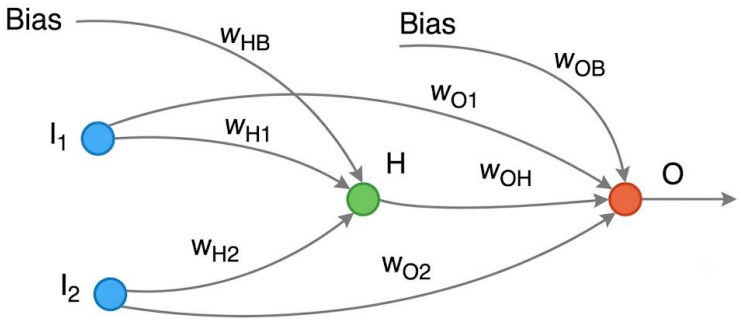
Karena minimisasi kesalahan mensyaratkan bahwa perubahan bobot harus searah dengan komponen gradien negatif, penyesuaian bobot untuk bobot ke-k dari i ditetapkan dengan mengalikan konstanta pembelajaran negatif:

$$\Delta w_{ki} = c * \partial Error/\partial w_{ki} = c * O_i(1 - O_i) * \sum_j (-delta_j * w_{ij}) x_k.$$

Untuk jaringan dengan lebih dari satu lapisan tersembunyi, prosedur yang sama diterapkan secara rekursif untuk menyebarkan kesalahan dari lapisan tersembunyi n ke lapisan tersembunyi $n - 1$. Meskipun memberikan solusi untuk masalah pembelajaran dalam jaringan multilayer, backpropagation bukannya tanpa kesulitan. Seperti algoritma hillclimbing, ia dapat konvergen ke minimum lokal, seperti yang terlihat pada Gambar 1.6 Terakhir, backpropagation dapat sangat mahal untuk dihitung, terutama ketika jaringan konvergen lambat.

Contoh: Backpropagation Memecahkan Masalah Exclusive-Or

Selanjutnya, kami mendemonstrasikan algoritma backpropagation untuk memecahkan masalah exclusive-or. Fungsi exclusive-or dalam logika menghasilkan nilai $true$ ketika salah satu dari dua nilai inputnya bernilai $true$, dan $false$ ketika kedua nilai inputnya bernilai $true$ atau $false$. Baru setelah terciptanya mesin Boltzmann, aturan delta umum, dan algoritma backpropagation, masalah exclusive-or terpecahkan. Gambar 1.9 menunjukkan jaringan dengan dua node input, satu node tersembunyi, dan satu node output. Jaringan ini juga memiliki dua node bias, yang pertama terhubung ke node tersembunyi dan yang kedua ke node output. Nilai bersih untuk node tersembunyi dan keluaran dihitung dengan cara biasa, sebagai hasil perkalian vektor dari nilai masukan dikalikan dengan bobot terlatihnya. Bias ditambahkan ke jumlah ini. Bobot dan bias dilatih menggunakan algoritma backpropagation dengan fungsi aktivasi sigmoid. Perhatikan bahwa node masukan juga dihubungkan langsung, dengan bobot terlatih, ke node keluaran. Penghubungan tambahan ini seringkali memungkinkan perancang jaringan untuk mendapatkan jaringan dengan lebih sedikit node di lapisan tersembunyi dan konvergensi yang lebih cepat.



Gambar 1.9 Sebuah jaringan saraf backpropagation yang menyelesaikan masalah exclusive-or. W_{ij} adalah bobot, dan I adalah node input, H adalah node tersembunyi, dan O adalah node output.

Tidak ada yang unik tentang jaringan pada Gambar 1.8 Sejumlah jaringan berbeda dapat digunakan untuk menghitung solusi untuk masalah exclusive-or. Jaringan yang diinisialisasi secara acak ini dilatih dengan beberapa contoh dari empat pola yang mewakili nilai kebenaran fungsi exclusive-or. Kami menggunakan simbol "→" untuk menunjukkan bahwa nilai fungsi tersebut adalah 0 atau 1. Keempat nilai ini adalah

$$(0,0) \rightarrow 0;\ (1,0) \rightarrow 1;\ (0,1) \rightarrow 1;\ (1,1) \rightarrow 0.$$

Sebanyak 1400 siklus pelatihan, menggunakan keempat contoh ini, menghasilkan nilai-nilai berikut, dibulatkan ke persepuluh terdekat, untuk parameter bobot pada Gambar 1.8 :

$$W_{H1} = -7.0;\ W_{H2} = 2.6;\ W_{HB} = -7.0;\ W_{O1} = -5.0;$$
$$W_{OH} = -11.0;\ W_{OB} = 7.0;\ W_{O2} = -4.0.$$

Dengan nilai input (0, 0), output dari node tersembunyi adalah

$$f(0 * (-7.0) + 0 * (-7.0) + 1 * (2.6)) = f(2.6) \rightarrow 1.$$

Output dari node output untuk (0, 0) adalah

$$f(0 * (-5.0) + 0 * (-4.0) + 1 * (-11.0) + 1 * (7.0)) = f(-4.0) \rightarrow 0.$$

Dengan nilai input (1, 0), output dari node tersembunyi adalah

$$f(1 * (-7.0) + 0 * (-7.0) + 1 * (2.6)) = f(-4.4) \rightarrow 0.$$

Output dari node output untuk (1, 0) adalah

$$f(1 * (-5.0) + 0 * (-4.0) + 0 * (-11.0) + 1 * (7.0)) = f(2.0) \rightarrow 1.$$

Nilai input (0, 1) serupa. Terakhir, kita periksa jaringan dengan nilai input (1, 1). Output dari node tersembunyi adalah

$$f(1 * (-7.0) + 1 * (-7.0) + 1 * 2.6) = f(-11.4) \rightarrow 0.$$

Output dari node output untuk (1, 1) adalah

$$f(1 * (-5.0) + 1 * (-4.0) + 0 * (-11.0) + 1 * (7.0)) = f(-2.0) \rightarrow 0.$$

Hasilnya menunjukkan bahwa jaringan feedforward pada Gambar 1.8 , menggunakan pembelajaran backpropagation, menghasilkan pemisahan nonlinier dari titik data exclusive-or. Fungsi ambang f adalah sigmoidal pada Gambar 7(c), bias yang dipelajari telah menggesernya sedikit ke arah positif sumbu x. Kami menawarkan representasi matriks yang menangkap bobot dan nilai input.

Sebagai kesimpulan dari contoh ini, penting untuk memahami apa yang dihasilkan oleh algoritma backpropagation. Ruang pencarian untuk jaringan exclusive-or kami memiliki delapan dimensi, yang diwakili oleh tujuh bobot pada Gambar 1.9 ditambah kesalahan output. Masing-masing dari tujuh bobot diinisialisasi dengan nilai acak. Ketika output awal dihasilkan dan kesalahannya ditentukan, backpropagation menyesuaikan masing-masing dari tujuh bobot untuk mengurangi kesalahan ini. Ketujuh bobot tersebut disesuaikan lagi dengan setiap iterasi algoritma, bergerak menuju nilai-nilai yang meminimalkan kesalahan untuk menghitung fungsi exclusive-or. Setelah 1400 iterasi, pencarian menemukan nilai untuk masing-masing dari tujuh bobot yang memungkinkan kesalahan mendekati nol. Yang terjadi adalah bahwa dalam ruang 8 dimensi, backpropagation telah menemukan hyperplane 7 dimensi yang secara tepat memisahkan keempat instance ex-or.

Akhirnya, sebuah pengamatan dibuat. Jaringan exclusive-or dilatih untuk memenuhi empat pola tepat, hasil dari penerapan fungsi exclusive-or pada pasangan benar/salah. Dalam situasi deep learning modern, pelatihan untuk menyelesaikan situasi tepat jarang terjadi. Ambil contoh, program yang memindai gambar sinar-X untuk mendeteksi situasi penyakit. Contoh lain adalah jaringan yang memindai las logam untuk mencari ikatan logam yang buruk. Sistem seperti itu disebut pengklasifikasi dan mereka memeriksa situasi baru yang belum pernah dilihat sebelumnya untuk menentukan apakah ada potensi masalah.

Aljabar Matriks

Selanjutnya kita mempertimbangkan bagaimana matriks digunakan untuk merepresentasikan beberapa contoh jaringan saraf yang telah kita lihat sebelumnya. Pertama-tama, perhatikan deskripsi umum neuron buatan yang dijelaskan pada Gambar 1.1. Di sini kita memiliki nilai input $x_1, x_2, x_3, \ldots, x_n$. Setiap nilai input ini memiliki bobot yang sesuai, $w_1, w_2, w_3, \ldots, w_n$. Perhitungan nilai node output adalah $\sum_i w_i x_i$ yang diberikan oleh hasil perkalian dua matriks:

$$[x_1, x_2, x_3, \dots, x_n] * \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = [\sum i\, w_i x_i].$$

Kemudian, fungsi ambang batas, f, diterapkan pada keluaran node: $f[\sum_i w_i x_i]$. Contoh kedua kami mempertimbangkan perhitungan bobot untuk jaringan pengkondisian stimulus Hebbian pada Bagian 1. Pertama, jaringan dijalankan dengan input dan bobot asli:

$$[x_1, x_2, x_3, \dots, x_6] * \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_6 \end{bmatrix} = [\sum i\, w_i x_i].$$

Penyesuaian bobot, ΔW, untuk iterasi kedua jaringan dibuat dengan mengalikan larik input asli dengan konstanta laju pembelajaran, c, dikalikan dengan output jaringan asli, yaitu 1 atau —1:

$$\Delta W = X * c * f(X, W).$$

Kumpulan bobot baru dihasilkan dengan perkalian skalar pada larik input asli:

$$[x_1 x_2 x_3 \dots x_6] * c * (1\ atau\ -1) = [2w_1\ 2w_2\ 2w_3 \dots 2w_6].$$

Selanjutnya, bobot baru ini dikalikan dengan larik transpos dari nilai input berikutnya, $[2x_1, 2x_2, 2x_3, \dots, 2x_6]$ untuk menghasilkan

$$[2w_1, 2w_2, 2w_3, \dots, 2w_6] * \begin{bmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \\ \vdots \\ 2x_6 \end{bmatrix} = [\sum i\, 2w_i 2x_i].$$

Output dari jaringan adalah tanda dari $\sum_i 2w_i 2x_i$, yaitu +1 atau —1. Penyesuaian bobot berikutnya, $[3w_1\ 3w_2\ 3w_3 \dots 3w_6]$, mirip dengan yang terbaru:

$$[2x_1\ 2x_2\ 2x_3 \dots 2x_6] * c * (1\ atau\ -1) = [3w_1\ 3w_2\ 3w_3 \dots 3w_6].$$

Jaringan tak terawasi Hebbian berlanjut, dengan detail numerik seperti yang ditunjukkan pada Bagian 1.1

Ukuran statistik

Pembelajaran mesin tradisional dalam AI selalu merupakan latihan dalam statistik komputasi. Misalnya, algoritma ID3 dan turunannya Bagian 10.3 menggunakan teori informasi untuk mengukur bagaimana potongan data dari kumpulan data yang besar berkorelasi. Analisis komponen utama sering digunakan untuk pengurangan dimensi matriks yang sangat besar yang digunakan dalam analisis semantik laten. Pengetahuan tentang kemungkinan paralelisme implisit dalam aljabar matriks telah menghasilkan efisiensi komputasi dalam pembelajaran mendalam.

Sebelumnya, kami telah menunjukkan bagaimana jaringan feedforward multi-lapisan menyebarkan kesalahan keluaran kembali ke lapisan tersembunyi jaringan. Penyebaran kesalahan ini disebut pembelajaran penurunan gradien. Kami juga mencatat bahwa node keluaran jaringan ini diaktifkan melalui penggunaan nilai-nilai yang dijumlahkan dari input berbobotnya. Selanjutnya, kami akan menjelaskan persamaan softmax yang mengubah nilai keluaran jaringan menjadi distribusi. Transformasi ini mengambil angka mentah yang dihasilkan oleh lapisan keluaran jaringan dan mengubahnya menjadi distribusi probabilitas. Istilah "*softmax*" berasal dari "pelunakan" fungsi maksimum tradisional yang memilih maksimum dari sekumpulan nilai yang diberikan.

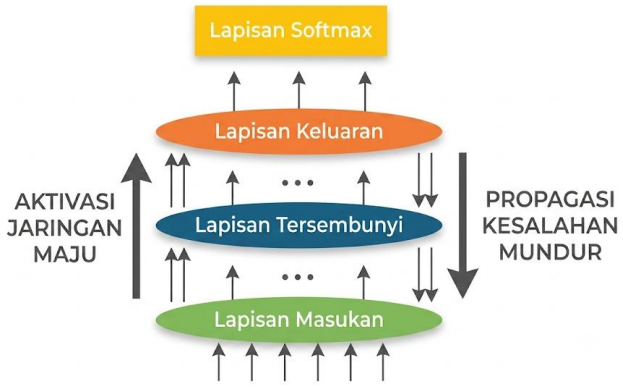
Seperti yang akan kita lihat di Bagian 1.3, algoritma pembelajaran mendalam merepresentasikan kata/tokennya sebagai bilangan riil kecil. Hal ini dilakukan karena berbagai alasan, termasuk menjaga turunan parsial dalam batas yang dapat diterima. Selanjutnya, karena output dari jaringan ini mencerminkan korelasi yang ditemukan antara kata/token dalam model yang dipelajari, penting untuk mengkarakterisasi nilai output dari banyak komputasi matriks sebagai distribusi probabilitas. Distribusi probabilitas adalah himpunan bilangan non-negatif yang jumlahnya sama dengan 1.0. Fungsi softmax s mendukung transformasi nilai output jaringan menjadi distribusi probabilitas:

$$\sigma(x_j) = e^{xj} / \sum_i e^{xi}.$$

Dalam rumus ini, Softmax dari setiap output x_j adalah e pangkat output tersebut dibagi dengan jumlah semua output sebagai pangkat e. Bahkan jika x_j adalah angka negatif, e pangkat tersebut tetap positif. Sebagai contoh, $s[1, 0, -1]$ kira-kira $[0{,}665, 0{,}244, 0{,}090]$. Perlu diketahui bahwa softmax mengubah setiap nilai output menjadi bagian dari distribusi probabilitas sambil mempertahankan hubungan keseluruhan antara nilai output asli. Gambar 10 melengkapi Gambar 1.5 dengan menambahkan lapisan Softmax. Hasil pemrosesan softmax kemudian digunakan dengan backpropagation untuk mengurangi kesalahan pada bobot node pada lapisan sebelumnya dari jaringan.

Gambar 1.10 menunjukkan softmax yang digunakan pada lapisan output jaringan, namun dapat digunakan pada nilai output dari kumpulan node di mana saja dalam jaringan. Salah satu contohnya akan ditunjukkan dengan komponen perhatian dari model transformer di Bagian 4. Ada beberapa masalah lebih lanjut yang muncul saat menggunakan softmax. Salah

satu masalahnya adalah mencoba memproses vektor besar dengan angka yang besar. Tetapi bahkan dengan vektor kecil, softmax dapat menghasilkan hasil yang berlebihan. Pertimbangkan, misalnya, Softmax dari vektor keluaran $[1\ 5\ 1\ 1]$. softmax menghasilkan vektor [0.0174 0.9479 0.0174 0.174], di mana elemen kedua dari vektor tersebut mendapatkan nilai yang sangat tinggi, hanya 0.0521 kurang dari 1, sementara probabilitas yang lebih kecil mendekati 0.



Gambar 1.10 Revisi Gambar 1.5 untuk menambahkan lapisan Softmax. Hasil Softmax kemudian digunakan untuk menyesuaikan bobot lapisan tersembunyi.

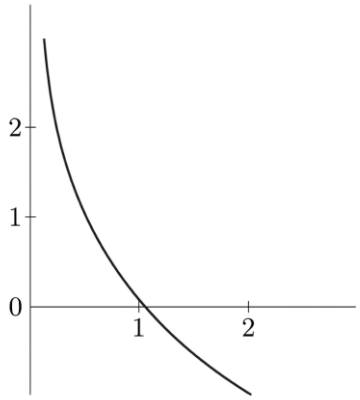
Masalah ini disebut saturasi dan hanya akan semakin buruk ketika beberapa angka jauh lebih besar daripada yang lain: hasil di mana angka yang lebih besar semakin mendekati 1 dan nilai yang lebih kecil mendekati 0. Masalah ini sangat penting karena vektor softmax yang dihasilkan dimaksudkan untuk digunakan untuk pengurangan kesalahan penurunan gradien, yang dapat menyebabkan konvergensi yang lambat dan varians yang tinggi dalam proses pelatihan. Ada solusi untuk masalah saturasi, seperti menormalisasi vektor keluaran dengan membagi setiap elemen dengan akar kuadrat dari panjang vektor, seperti yang akan kita lihat di bagian selanjutnya.

Terakhir, pembuatan jaringan berdasarkan penalaran menggunakan distribusi probabilitas memengaruhi seluruh desain jaringan. Embedding kata/token yang dimasukkan ke jaringan direpresentasikan sebagai bilangan riil kecil, atau float, antara, dan termasuk, satu dan nol. Vektor bobot diinisialisasi secara acak sebagai bilangan riil mendekati nol. Kita akan melihat hal ini secara lebih detail dengan model bahasa besar dan transformer pada Bagian 1.3, Setiap output dari lapisan output jaringan saraf, sebelum transformasi softmax-nya, disebut logit, diucapkan "LOW-jit". Menggunakan distribusi probabilitas untuk menentukan ukuran hadiah/hukuman untuk bobot dalam jaringan disebut menghitung kerugian entropi silang. Entropi adalah istilah dari teori informasi yang mengukur tingkat kekacauan atau keacakan dalam suatu sistem. Kehilangan entropi menunjukkan pengurangan kekacauan atau kesalahan. Ketika persamaan Softmax digunakan, pengurangan kesalahan ini diukur secara probabilistik di seluruh keadaan output jaringan. Entropi silang adalah istilah yang digunakan untuk menggambarkan perbedaan antara dua distribusi, yaitu distribusi output saat ini dan output yang diinginkan.

Penurunan kerugian entropi silang, sebuah angka positif, diukur menggunakan probabilitas logaritmik negatif yang diberikan pada output Softmax. Softmax menghitung probabilitas semua output dan memberikan urutan hasil "terbaik" untuk meminimalkan kerugian. Kita menyatakan kerugian entropi silang, CEL, x, sebagai probabilitas logaritmik negatif dari hasil tersebut:

$$CEL(f, x) \; = \; - ln \, p_f \, (a_x).$$

Dalam persamaan ini, (f, x) mewakili persamaan Softmax, yang diterapkan pada x yang menghasilkan a_x. Selanjutnya kita mempertimbangkan mengapa probabilitas logaritmik negatif digunakan. Pertama, estimasi kesalahan kita dari softmax adalah angka positif sehingga kita ingin mengurangi kesalahan tersebut. Kedua, fungsi logaritmik memiliki sifat penting antara 1, probabilitas maksimum kita, dan 0: — ln(x) menjadi lebih besar saat x mendekati 0, yang penting karena rentang inilah yang akan menjadi fokus utama kita. Fungsi logaritmik x ini ditunjukkan pada Gambar 1.11.



Gambar 1.11 Grafik — ln(x) saat x mendekati 0.

1.3 MEMBANGUN MODEL BAHASA YANG BESAR

Pertimbangkan kalimat "Apa yang dimodelkan oleh model bahasa?" Berapa kali dalam komunikasi bahasa Inggris normal kata model mengikutinya sendiri? Tidak terlalu sering. Di sisi lain, berapa kali kata "tangan" akan mengikuti tiga kata "di sisi lain"? Cukup teratur, karena membuat perbandingan menggunakan metafora biner cukup umum dalam komunikasi bahasa Inggris normal. Kedua situasi ini ditangkap dalam model bahasa. Model bahasa dirancang untuk mewakili bagaimana bahasa tertentu digunakan. Model bahasa saat ini yang dikembangkan untuk jaringan saraf dalam bersifat stokastik, di mana setiap kueri ke model menghasilkan distribusi probabilitas pola bahasa yang memenuhi kueri tersebut. Data untuk mendukung model ini dapat berasal dari penggunaan bahasa tertulis atau lisan. Model bahasa yang besar adalah representasi probabilistik dari korpus penggunaan bahasa yang digunakan untuk melatihnya. Selanjutnya kita akan fokus pada pembangunan model untuk komunikasi kata/kalimat bahasa Inggris tertulis.

Seperti yang akan kita lihat, menemukan kata apa yang seharusnya menjadi kata berikutnya dalam suatu ekspresi penting untuk menciptakan ekspresi baru. Bahkan, model bahasa juga mendukung pencarian Kata yang paling mungkin sebelumnya untuk sekumpulan kata, dan kata-kata yang paling mungkin cocok di antara rangkaian bahasa. Proses pembangkitan kata yang hilang ini mendukung desain dan penggunaan banyak aplikasi berbasis bahasa yang menarik. Ini dapat mencakup:

A. Penerjemahan antar bahasa.
B. Klasifikasi teks ke dalam berbagai kategori.
C. Pengenalan "sentimen" seperti kegembiraan atau kemarahan dalam ekspresi.
D. Pengambilan informasi dari basis data.
E. Menjawab pertanyaan, seperti dalam berbagai tes standar.
F. Pembuatan artikel berita.
G. Menyusun esai tentang berbagai topik.

Ada beberapa komponen dalam pembuatan model bahasa Inggris tertulis yang besar. Kita akan membahasnya selanjutnya.

Mempersiapkan teks

Merepresentasikan teks berbahasa Inggris sepenuhnya, bahkan untuk jaringan saraf yang sangat besar, adalah tugas yang kompleks. Pertama, ada lebih dari jumlah angka yang tak terhingga, yang masing-masing dapat menjadi bagian dari sebuah teks. Tidak ada mesin keadaan terbatas yang dapat menangani jumlah token yang tak terhingga! Dengan demikian, tugas pertama dalam merepresentasikan bahasa tertulis untuk jaringan saraf adalah memecah teks menjadi sejumlah bagian yang terbatas dan masuk akal, yang akan kita sebut sebagai "token". Kemudian kita akan menjelaskan bagaimana token-token ini akan menjadi masukan untuk jaringan. Kita dapat mulai dengan mempertimbangkan contoh jumlah angka yang tak terhingga dan mengganti setiap angka tersebut dengan token UNK. Token ini, yang menunjukkan kata "tidak dikenal" dalam bahasa Inggris, juga akan digunakan untuk mengganti komponen teks apa pun yang tidak ada dalam kumpulan token kosakata akhir yang akan menjadi masukan ke jaringan.

Karena bahkan jaringan saraf yang sangat besar pun memiliki ukuran tetap tertentu, selanjutnya kita mempertimbangkan berapa jumlah token kosakata yang "dapat diterima". Model bahasa saat ini memiliki kosakata kata sebanyak 30.000, dengan BERT, hingga 50.000 untuk GPT. Meskipun ini terdengar seperti jumlah token yang cukup besar untuk jaringan, Oxford English Dictionary memperkirakan ada lebih dari 170.000 kata bahasa Inggris yang saat ini digunakan. OED juga berisi sekitar 50.000 kata usang yang digunakan pada waktu yang berbeda. GPT, saat ini, dilatih pada korpus kalimat bahasa Inggris yang berisi lebih dari setengah triliun kata dari 45 terabyte data teks. Oleh karena itu, tugasnya, setelah korpus pelatihan diidentifikasi, adalah mengurangi jumlah total kata individual dalam korpus pelatihan menjadi kumpulan token spesifik yang dapat diproses oleh jaringan. Ada beberapa metode untuk menyelesaikan tugas ini:

(A) Kata-kata individual harus diisolasi dan "dikenali". Isolasi ini dapat dibantu dengan mengenali spasi kosong yang memisahkan kata-kata. Tanda baca, seperti titik atau titik

koma, juga dapat memisahkan kata-kata. "Pengenalan" kata ini memungkinkan penyematan kata, yang dijelaskan di bawah ini.

(B) Simbol tanda baca "dikenali" menggunakan aturan yang menjelaskan peran mereka dalam teks bahasa Inggris. Simbol tanda baca itu sendiri ditempatkan dalam kategori "UNK".

(C) Kalimat biasanya diberi tambahan, dengan menambahkan simbol tertentu, misalnya, "STOP" setelah setiap kalimat. STOP ini menangkap perbedaan antara titik yang mengakhiri kalimat dan titik yang merupakan bagian dari kalimat, misalnya, misalnya.

(D) Paragraf baru, halaman baru, kata yang salah eja dan kata yang diberi tanda hubung juga harus dikenali dan ditokenisasi, mungkin sebagai UNK.

(E) Dan seterusnya...

Seperti yang dapat dipahami dengan sedikit berpikir, proses pembuatan aturan untuk memecah teks menjadi kata/token individual ini kompleks dan juga membosankan untuk diterapkan. Namun, komputer sangat unggul dalam tugas-tugas yang kompleks dan membosankan. Detail lebih lanjut tentang proses tokenisasi teks bahasa Inggris dapat ditemukan dalam teks linguistik komputasional atau di internet.

Tugas selanjutnya adalah menentukan jumlah kata atau token unik, yaitu kosakata V, untuk model bahasa. Kami juga menghitung berapa kali setiap kata/token ini digunakan dalam korpus pelatihan. Setelah kata-kata dikatalogkan dan kemunculannya dihitung, pengurangan lebih lanjut merupakan fungsi dari keterbatasan kosakata dalam jaringan. Pengurangan dilakukan dengan beberapa cara, termasuk menghapus simbol non-kata dari teks. Selain angka, titik, koma, dan tanda baca lainnya juga dihapus. Ekspresi kimia, biologi, atau matematika juga sering dihapus. Kata-kata yang jarang dan tidak sering digunakan juga dapat dihapus.

Sekitar tahun 2017 dan dengan munculnya GPT-1, penyempurnaan lebih lanjut dilakukan pada tokenisasi bahasa. Semua kata, termasuk UNK sebelumnya, diuraikan lebih lanjut untuk membuat token kata parsial. Proses ini memecah semua kata menjadi suku kata penyusunnya atau, jika itu adalah ucapan bersuara, menjadi fonem. Proses ini masuk akal karena jumlah suku kata atau fonem jauh lebih sedikit daripada jumlah kata individual dalam suatu bahasa. GPT sekarang menggunakan sekitar 50.000 token bahasa ini. Setelah ukuran himpunan token yang sesuai dengan batasan jaringan, V, ditentukan, pemetaan dari token ke representasi jaringan dimungkinkan. Ini disebut membuat *embedding* kata atau token.

Membuat embedding kata/token

Karena model bahasa akan direpresentasikan sebagai distribusi probabilitas atas penggunaan kata/token dalam suatu bahasa, pengkodean kata/token, yang dihasilkan secara acak, biasanya diambil dari himpunan bilangan riil antara -1 dan 1. Setiap token akan direpresentasikan sebagai vektor bilangan riil, atau float, yang disebut embedding kata. Jumlah float, f, yang digunakan untuk embedding kata adalah hyperparameter jaringan. Nilai tipikal untuk f adalah 100, meskipun vektor yang jauh lebih besar sering digunakan untuk embedding token. Ukuran array embedding token adalah $|V| \times f$, di mana $|V|$ adalah ukuran kosakata dan f adalah jumlah float yang digunakan dalam model. Bilangan bulat 1 hingga $|V|$ berfungsi

sebagai indeks untuk array embedding token. Jadi, jika token "mentega" memiliki indeks 6, maka baris ke-6 dari array embedding adalah vektor yang mewakili "mentega". Array embedding kata ini merupakan parameter dari jaringan pembelajaran. Ketika jaringan dilatih pada pola kata/token bahasa Inggris dan, misalnya, mencari kata/token berikutnya yang paling tepat dalam rangkaian kata/token, back propagation mengkondisikan semua embedding kata.

Lebih tepatnya, input ke model bahasa adalah indeks kata/token yang sedang diteliti yang langsung diterjemahkan ke dalam embedding-nya. Melanjutkan dari titik itu, semua operasi jaringan dilakukan pada embedding. Output jaringan adalah probabilitas bahwa setiap token dalam array embedding adalah token berikutnya yang paling mungkin dalam suatu ekspresi. Pengkondisian backpropagation adalah kerugian cross-entropy, yang dijelaskan oleh fungsi — ln $P\ (x)$ pada Gambar 1.7 Ini adalah logaritma natural negatif dari probabilitas x, di mana x adalah contoh kata berikutnya yang benar. Pada tahap backpropagation, setelah fungsi kerugian ditentukan, embedding token dari kerugian tersebut dimodifikasi.

Bi-gram dan tri-gram

Teknologi N-gram telah lama menjadi bagian penting dari pemahaman bahasa alami dan pengembangan model bahasa. Pertanyaan umum menanyakan berapa probabilitas suatu ekspresi menjadi bagian dari suatu bahasa. Dalam istilah probabilistik, yang dijelaskan lebih rinci dalam BAGIAN VII, probabilitas, P, dari ekspresi "Hari ini panas" dapat direpresentasikan menggunakan aturan rantai:

$$P\ (\text{Hari ini panas})\ =\ P\ (Itu) * P\ (\text{adalah} \mid Itu) * P\ (\text{panas} \mid \text{Itu}) * P\ (\text{hari ini} \mid \text{Itu panas}).$$

Ekspresi $P\ (X|Y\ )$ diinterpretasikan sebagai "probabilitas bahwa X benar dengan syarat Y benar". “Hari ini panas” jelas merupakan kalimat yang sangat pendek. Tidak sulit membayangkan perhitungan dan pelatihan jaringan yang diperlukan untuk menentukan probabilitas ekspresi yang lebih panjang, misalnya 15 kata. Intinya, selain perhitungan probabilitas itu sendiri, adalah kurangnya korpus yang cukup untuk mengkondisikan semua probabilitas yang membentuk anggota dari hasil perkalian probabilitas ini merupakan batasan serius pada pembelajaran. Asumsi Markov menyederhanakan perhitungan yang panjang ini, dan membatasi probabilitas yang membentuk rantai, dengan mengasumsikan bahwa probabilitas suatu kata hanya dikondisikan oleh probabilitas kata yang mendahuluinya. Contoh kita sebelumnya, yang disederhanakan dengan menggunakan asumsi Markov, adalah:

$$P\ (\text{Hari ini panas})\ =\ P\ (\text{Itu}) * P\ (\text{adalah} \mid \text{Itu}) * P\ (\text{panas} \mid \text{adalah}) * P\ (\text{hari ini} \mid \text{panas}).$$

Penggunaan asumsi Markov untuk menentukan probabilitas kombinasi kata disebut bi-gram, atau model dua kata. Probabilitas $bi-gram$ ini dapat dikembangkan ke dua arah: mengambil sebuah kata dan menentukan kata berikutnya yang paling mungkin, atau mengambil sebuah kata dan menentukan probabilitas kata sebelumnya. Kita dapat memahami bagaimana penentuan probabilitas bi-gram dari pasangan kata sangat penting untuk menghasilkan urutan pola kata yang koheren. Probabilitas tri-gram bahkan lebih ampuh dalam menangkap pola

penggunaan bahasa. $Tri-gram$, atau model tiga kata, menggambarkan probabilitas sebuah kata sebagai fungsi dari probabilitas dua kata yang mendahuluinya atau dua kata yang mengikutinya. Dibutuhkan korpus bahasa yang lebih besar untuk melatih model tri-gram daripada yang dibutuhkan untuk melatih model bi-gram.

Melatih model bahasa

Banyak detail yang membentuk generasi model bahasa besar saat ini tetap menjadi rahasia perusahaan. Dibutuhkan pencarian internet dan komunikasi pribadi untuk menentukan perkiraan terbaik tentang bagaimana model-model ini disusun dan dilatih. Namun, ada informasi yang menjelaskan waktu dan biaya pelatihan LLM: Misalnya, Google membutuhkan waktu empat hari untuk 64 prosesor tensor yang dikonfigurasi khusus untuk melatih BERT. Untuk GPT-3 yang lebih umum, diperkirakan pelatihan membutuhkan waktu sekitar 34 hari untuk 1.024 komputer A100 dengan unit pemrosesan grafis. Diperkirakan juga bahwa biaya pelatihan GPT-3 menggunakan penyedia cloud GPU dengan biaya terendah adalah sekitar Rp46 Miliar. Detail tentang Gemini, LLM DeepMind milik Google, tidak tersedia.

Untungnya bagi pengguna akhir, setelah model-model ini dilatih, model tersebut dapat digunakan kembali dengan pelatihan ulang minimal oleh kelompok luar untuk kebutuhan khusus mereka sendiri. Idenya adalah bahwa lapisan terakhir jaringan dapat diganti, dan pola yang dipelajari dari lapisan tersembunyi dapat dikondisikan ulang untuk menangani tugas-tugas baru. Pelatihan baru ini dapat memakan waktu sekitar 2 jam pada laptop biasa menggunakan prosesor grafis. Akibatnya, waktu yang dibutuhkan pengguna umum untuk menggunakan lingkungan BERT atau GPT untuk penggunaan khusus jauh lebih sedikit. Kami menjelaskan penggunaan ulang, atau penyempurnaan, LLM ini secara lebih rinci di Bagian 5.

Tabel 1.3 Hasil yang dirangkum, menggunakan model bahasa berbasis bi-gram.

Kata	Kemiripan Kosinus
itu	-0.160
sebuah	0.127
mengingat	0.479
mengatakan	0.553
komputer	0.249
mesin	0.333

Menarik untuk mempertimbangkan beberapa komputasi yang dihasilkan oleh model bahasa. Charniak di Bab 4 menjelaskan hasil menjalankan model tersebut. Model bahasanya dilatih pada korpus sekitar 1 juta kata dan memiliki ukuran kosakata 7500 kata. Panjang embedding kata adalah 30. Hasil terpilih dari model Charniak disajikan dalam Tabel 1.3 Kesamaan kosinus mengukur vektor embedding kata dengan panjang 30. Perhatikan bahwa kata-kata yang digunakan dalam situasi serupa memiliki nilai kosinus yang berdekatan. Menariknya, kata-kata yang sesuai untuk dimasukkan ke dalam urutan kata juga dapat memiliki arti yang serupa. Misalnya, mengingat atau mengatakan masing-masing dapat mengikuti dia atau dia dari nama seseorang dalam sebuah kalimat. Selanjutnya kita mempertimbangkan desain model bahasa besar berbasis perhatian yang disebut transformer.

1.4 MENUJU MODEL BAHASA BESAR BERBASIS TRANSFORMER

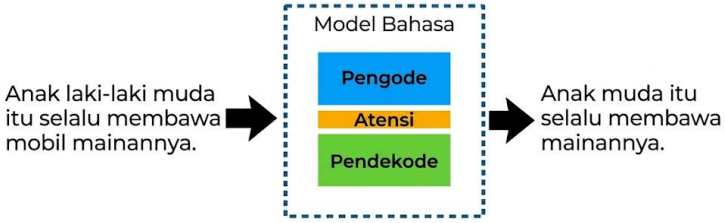
Pendahulu model bahasa berbasis transformer, sebelum rilis GPT-1 pada tahun 2018, adalah jaringan saraf konvolusional dan rekuren serta autoencoder. Jaringan konvolusional adalah jaringan feedforward yang mempelajari fitur gambar melalui optimasi filter. Jaringan rekuren mendukung umpan balik di dalam lapisan sehingga input baru untuk beberapa node dimodulasi oleh output node sebelumnya. “Perangkat memori” ini memfasilitasi penggunaannya dalam tugas pengenalan tulisan tangan dan ucapan otomatis. Autoencoder adalah jaringan tanpa pengawasan untuk mempelajari representasi efisien dari data yang tidak berlabel. Elemen decoder kemudian menerjemahkan representasi internal ini kembali ke dalam beberapa bentuk yang berguna. Misalnya, autoencoder dapat bersifat generatif, menerjemahkan rangkaian kata ke dalam bahasa yang berbeda.

Arsitektur yang mendasari jaringan konvolusional dan rekuren adalah implementasi yang ditingkatkan dari ide-ide beberapa dekade yang lebih tua. Model yang menyerupai jaringan saraf konvolusional klasik memperoleh status mutakhir dalam visi komputer dan model yang menyerupai desain asli untuk jaringan saraf rekuren LSTM mendominasi aplikasi dalam pemrosesan bahasa alami. Sepanjang tahun 2010-an, munculnya keberhasilan pembelajaran mendalam yang cepat berdasarkan teknologi CNN dan RNN juga disebabkan oleh ketersediaan sumber daya komputasi, termasuk inovasi dalam komputasi paralel dan pusat data besar, yaitu penyimpanan murah dan beberapa penyedia layanan internet.

Namun, ada keterbatasan mendasar pada RNN termasuk penggunaan LSTM Jaringan berulang. Pertama, jaringan berulang mengalami kesulitan dalam menangani ketergantungan jarak jauh dalam kalimat. Kedua, RNN memiliki masalah dengan gradien pembelajaran yang sangat besar dan sangat kecil. Terakhir, sifat sekuensial dari "*roll out*" jaringan berulang membatasi evaluasi paralel dan membutuhkan waktu pelatihan yang jauh lebih lama. Model berbasis perhatian mengatasi masing-masing masalah ini. Satu kontribusi lebih lanjut dari awal tahun 2010-an, dan langkah penting dalam evolusi model bahasa berbasis transformer, adalah penciptaan jaringan adversarial generatif, atau GAN. GAN awalnya dikembangkan oleh Goodfellow dan rekan-rekannya. GAN mengadu dua jaringan saraf "melawan" satu sama lain, komponen adversarial, dalam kompetisi zero-sum. Dengan diberikan set pelatihan, teknik GAN dapat menghasilkan data baru dengan profil statistik yang sama dengan data pelatihannya. Misalnya, GAN yang dilatih pada data gambar menghasilkan gambar baru yang dapat terlihat autentik bagi penonton manusia (url 1). Meskipun awalnya diusulkan sebagai metode untuk pembelajaran tanpa pengawasan, GAN juga berguna untuk pembelajaran semi-supervised dan. Ia juga dapat mendukung pembelajaran penguatan.

Ketika pertama kali diusulkan, mekanisme perhatian menghasilkan peningkatan pada jaringan berulang yang sebelumnya digunakan untuk penerjemahan mesin, berkinerja lebih baik daripada pendekatan sekuens-ke-sekuens encoder-decoder sebelumnya. Dengan menggunakan perhatian, decoder menerima sebagai input vektor konteks yang terdiri dari representasi berbobot dari input pada setiap langkah waktu. Para peneliti mencatat bahwa beberapa wawasan kualitatif penting sering muncul ketika menggunakan bobot perhatian.

Dalam tugas penerjemahan, misalnya, perhatian dapat menyarankan pilihan antara sinonim untuk menghasilkan kata-kata dalam bahasa target. Pada Gambar 1.12, perhatian meningkatkan model autoencoder-decoder.



Gambar 1.12 Perhatian Bahdanau memediasi antara input ke encoder dan output decoder dalam contoh terjemahan bahasa Inggris ke bahasa Prancis.

Dimulai dengan GPT-1 OpenAI, yang tersedia pada tahun 2018, transformer telah menjadi arsitektur jaringan yang dominan untuk AI generatif. Hampir semua tugas pemrosesan bahasa dan analisis visi saat ini didasarkan pada arsitektur transformer. Selain GPT, contohnya termasuk BERT, RoBERTa, Longformer, dan Gemini. Model bahasa besar OpenAI dapat mengembangkan "percakapan" menggunakan transformer GPT. Lebih lanjut, transformer visi mereka muncul sebagai model untuk berbagai tugas visi, termasuk pengenalan gambar dan deteksi objek, serta menciptakan dukungan semantik. Transformer juga digunakan untuk pengenalan ucapan dan pembelajaran penguatan. Model transformer menggunakan autoencoder yang berisi mekanisme self-attention. Kita akan membahasnya selanjutnya.

Model Transformer

Artikel berjudul *Attention is All You Need* merupakan kontribusi revolusioner terhadap desain model generatif. Ide yang mendukung desain transformer adalah mekanisme *self-attention*. Gambar 1.12, diadaptasi dari, mencerminkan model transformer dengan mekanisme perhatian, yang disebut self-attention, di setiap komponen *encoder–decoder*. Mekanisme perhatian meningkatkan model transformer karena mendukung pemfokusan selektif pada beberapa elemen input. Hal ini meningkatkan akurasi decoding dan efisiensi komputasi. Mekanisme perhatian memprioritaskan dan menekankan informasi yang relevan, bertindak sebagai "fokus" yang meningkatkan kinerja model secara keseluruhan. Ini mengatasi salah satu masalah model rekuren sebelumnya di mana ketergantungan jarak jauh dalam satu kalimat atau di beberapa kalimat sering hilang.

Seperti yang terlihat pada Tabel 1.4, saat ini terdapat beberapa model bahasa besar berbasis transformer yang tersedia untuk dieksplorasi. Banyak detail transformer ini tidak tersedia untuk umum. Kami akan membatasi komentar kami pada publikasi yang tersedia. Secara khusus, kami fokus pada arsitektur transformer asli yang diusulkan oleh Vaswani dan rekan-rekan, seperti yang terlihat pada Gambar 1.13, dan pada deskripsi transformer yang ditawarkan oleh Zhang dan rekan-rekannya. Encoder memetakan urutan input embedding token, $(x_1, x_2, \ldots, x_n)$ ke urutan representasi kontinu $(z_1, z_2, \ldots, z_n)$. Dengan mengonsumsi Z, decoder kemudian menghasilkan, satu per satu, urutan output $(y_1, y_2, \ldots, y_m)$. Pada setiap Pada langkah selanjutnya, saat menghasilkan output, dekoder dapat memanfaatkan semua

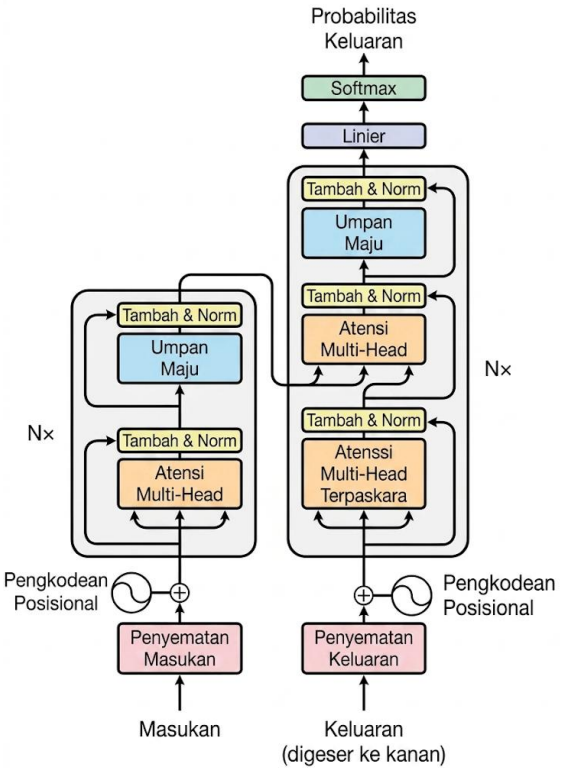
simbol yang telah dihasilkan sebelumnya sebagai input tambahan. Properti ini disebut autoregresif.

Encoder untuk transformer asli adalah tumpukan enam lapisan identik. Seperti yang terlihat pada Gambar 1.12, terdapat dua komponen di setiap lapisan, yang pertama adalah mekanisme self-attention multi-head dan yang kedua adalah jaringan perceptron feed-forward yang terhubung penuh. Terdapat koneksi add dan norm untuk setiap komponen untuk memastikan komunikasi antar komponen dan antar lapisan. Dekoder asli juga merupakan tumpukan enam lapisan identik. Selain dua subkomponen yang identik dengan encoder, terdapat mekanisme multi-head attention kedua, masked, dan positional encoding yang memastikan bahwa embedding output diimbangi satu posisi. Ini memastikan bahwa nilai output pada setiap saat juga bergantung pada nilai output sebelumnya. Komponen dekoder juga memiliki mekanisme add dan norm yang mendukung komunikasi antar komponen dan antar lapisan.

Tabel 1.4 Saat ini, Juli 2024, perangkat lunak yang tersedia untuk AI generatif. Perhatikan bahwa informasi tentang beberapa model Informasi ini bersifat rahasia perusahaan. Pembaca yang tertarik sebaiknya mencari informasi lebih lanjut mengenai perangkat lunak ini karena perusahaan AI diketahui sering berganti nama, bergabung, atau bubar. Tabel diadaptasi dari.

Model	Kemampuan	Parameter	Data pelatihan	URL
BERT	Menjawab pertanyaan, menemukan kemiripan semantik	345 juta	3,3 miliar kata	url 2
PaLM 2	Menghasilkan teks, esai, dan laporan, menjawab pertanyaan, menggunakan gaya dan nada yang diinginkan. Disetel untuk mengikuti instruksi	340 miliar	3,6 triliun token	url 3
ChatGPT, didukung oleh GPT	Menghasilkan teks, esai, dan laporan, menjawab pertanyaan, menggunakan gaya dan nada yang diinginkan. Disetel untuk mengikuti instruksi.	175 miliar	300 miliar token	url 4
Gemini dibuat oleh Google DeepMind	Menghasilkan keluaran multimodal dari masukan multimodal. Menghasilkan dokumen dengan teks dan gambar.	Informasi tidak publik	Informasi tidak publik	url 16 url 17
Llama 2	Menghasilkan teks, esai, dan laporan, menjawab pertanyaan menggunakan gaya dan nada yang diinginkan.	70 miliar	2 triliun token	url 5
T5	Pembuatan teks multibahasa tingkat lanjut	220 juta	29 triliun karakter dalam 107 bahasa	url 6
Stable diffusion	Menghasilkan gambar dari teks, terjemahan gambar.	860 juta	5 miliar pasangan anotasi teks gambar	url 7

Imagen	Menghasilkan gambar dari teks, terjemahan gambar, pengeditan gambar tingkat lanjut.	Informasi tidak publik	Informasi tidak dirilis ke publik	url 8
DALL E	Menghasilkan gambar dari teks, modifikasi gambar tingkat lanjut	12 miliar	650 juta pasangan teks gambar	url 9
Phenaki	Menghasilkan video dari deskripsi teks	Informasi tidak publik	Informasi tidak dirilis ke publik	url 10
Music Gen	Menghasilkan musik dari deskripsi teks	Informasi tidak publik	Informasi tidak dirilis ke publik	url 11



Gambar 1.13 Model transformer, yang mencerminkan encoder dan decoder menggunakan self-attention. Ini adalah satu lapisan dari arsitektur tumpukan enam lapis asli.

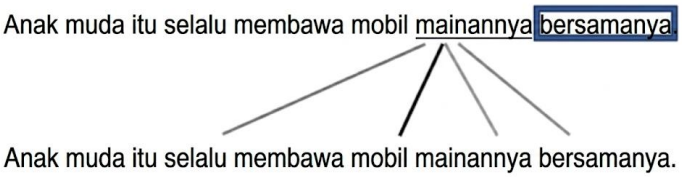
Perhatian

Gambar 1.12 menyajikan contoh di mana encoder dan decoder berfokus pada serangkaian token untuk menerjemahkan rangkaian tersebut ke dalam bahasa Prancis. Kita mulai bagian ini dengan dua contoh mekanisme self-attention yang mendukung pemrosesan transformer. Dalam transformer, self-attention adalah fungsi jaringan yang membantu encoder dan decoder dalam menentukan bagaimana token dalam serangkaian token saling terkait; contohnya ditunjukkan pada Gambar 1.14 Pada kedua gambar tersebut, rangkaian di bawah identik dengan rangkaian di atas dan perhatian harus dilihat sebagai pencarian hubungan di dalam rangkaian itu sendiri. Pada Gambar 1.14, fokusnya adalah pada token "it", mencari referensi yang sesuai; di sebelah kiri perhatian berfokus pada "animal" dan di sebelah kanan pada "street".



Gambar 1.14 Perhatian: menemukan token yang relevan yang mendukung penggunaan kata ganti "it" dalam dua kalimat berbeda tetapi berhubungan secara sintaksis.

Sebagai contoh kedua, pertimbangkan kalimat pada Gambar 1.15, "*The young boy always carries his toy car with him.*" Jika kata "*him*" disembunyikan, dan model tidak mengingat "*boy*" dari awal kalimat, model tidak akan tahu kata ganti mana yang harus digunakan di akhir kalimat: "*him*", "*her*", atau "*it*". Memperluas contoh ini ke tugas penerjemahan, transformer perlu menggunakan kata ganti yang diterjemahkan dengan tepat ketika merujuk pada anak laki-laki.



Gambar 1.15 Perhatian diri untuk his. Arsir garis menunjukkan jumlah perhatian yang diberikan kata his kepada kata-kata lain dalam urutan tersebut. Dengan menggunakan perhatian terselubung, atau perhatian autoregresif, hanya kata-kata sebelum kata yang sedang dipertimbangkan yang dikunjungi, dalam contoh ini, tiga baris paling kiri. Gambar diadaptasi dari.

Untuk mendefinisikan mekanisme perhatian, kita mempertimbangkan tiga komponen. Pertama, sebuah kueri, q, yang dapat menyelidiki tensor, T, dari m tupel, untuk informasi spesifik. Penyelidikan ini menanyakan lokasi tensor, atau kunci, untuk mengakses informasi di lokasi spesifik tersebut. Lokasi tensor yang kita sebut sebagai kunci, k, dan isi dari lokasi tersebut sebagai nilai v, fokus dari kueri. Kueri, kunci, nilai, dan output semuanya adalah tensor. Perhatian dan perhatian diri didefinisikan sebagai:

$$\text{Perhatian}(a, T) = \sum_i a(q, k_i)v_i,$$

di mana q adalah token kueri, i mencakup n pasangan kunci-nilai dari tensor T dan a adalah bobot perhatian berupa bilangan riil positif. Dengan self-attention, tensor T berisi kueri, q, kunci k, dan nilai v, yaitu perhatian difokuskan pada vektor yang berisi kueri. Untuk meringkas hasil penggunaan mekanisme perhatian:

(1) Proses kueri perhatian, q, dapat beroperasi pada tensor, T, dengan ukuran apa pun.
(2) Satu kueri, q, akan menerima respons yang berbeda, v, tergantung pada kunci tensor, T.
(3) Kueri, q, yang beroperasi pada T dapat fleksibel, yaitu meminta kecocokan yang tepat atau perkiraan.
(4) Bobot a pada kueri adalah bilangan riil positif. Proses perhatian membentuk kerucut positif, C. Artinya, jika v_i dan v_j adalah keluaran dari prosedur perhatian dan c dan d adalah skalar positif, maka semua $cv_i + dv_j$ juga termasuk dalam kerucut C.

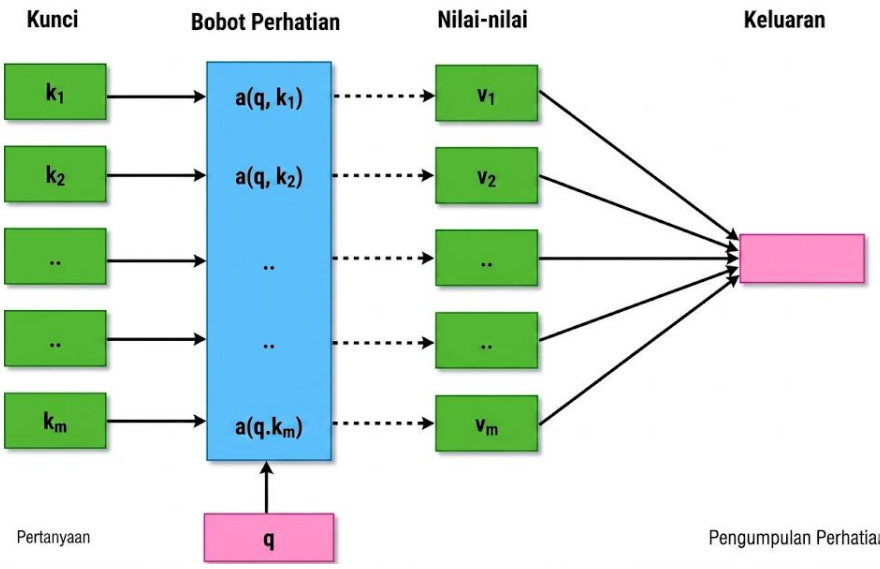
Untuk memastikan bahwa semua bobot perhatian tidak negatif dan jumlahnya sama dengan 1, kita menerapkan bentuk fungsi *softmax*:

$$a(q, k_j) = \frac{e^{a(q,k_j)}}{\sum_i e^{a(q,k_i)}}.$$

Perhatian mendukung agregasi, atau penggabungan, informasi, berdasarkan kueri q, di atas beberapa pasangan kunci-nilai. Prosedur perhatian itu sendiri menyediakan fungsi kontinu dan terdiferensiasi untuk jaringan saraf feed-forward untuk menentukan elemen mana yang paling sesuai untuk konstruksi representasi berbobot lebih lanjut. Bobot perhatian yang melekat pada setiap pasangan kueri-kunci dilatih oleh nilai-nilai yang diakses oleh kunci dalam pasangan kunci-kueri. Gambar 1.16 menawarkan contoh pemrosesan jaringan penggabungan perhatian.

Selanjutnya, kami menjelaskan beberapa metrik yang mendukung analisis kueri-kunci, atau bagaimana hubungan kunci-nilai berbasis kueri, $a(q, k_i)$ dapat dilatih untuk setiap v_i. Salah satu metrik yang umum digunakan adalah Gaussian. Sebagai contoh penggunaan metrik Gaussian, estimator Nadaraya–Watson digunakan dalam analisis regresi di mana kueri mewakili lokasi untuk melakukan regresi, kunci adalah lokasi data sebelumnya yang diamati, dan nilai sesuai dengan nilai regresi. Untuk contoh regresi Gaussian, ukuran $a(q, k)$ dihitung:

$$a(q, k_i) = e^{(-12|q-k_i|^2)}$$



Gambar 1.16 Prosedur perhatian, dengan pooling, menghitung fungsi kontinu dan terdiferensiasi dari nilai bobot di atas v_i. Bobot dilatih untuk (q, k_i) dengan nilai v_j.

di mana $||q - k_i||$ adalah perbedaan vektor ternormalisasi antara q dan setiap kunci, k_i. Terdapat pilihan alternatif untuk menentukan $a(q, k_i)$. Yang paling umum adalah produk titik berskala yang digunakan dalam transformer asli Vaswani dan rekan-rekan. Produk titik adalah nilai skalar yang dibuat dengan perkalian komponen demi komponen dari dua vektor dan kemudian penjumlahan dari produk-produk ini. Untuk memastikan bahwa besarnya produk titik tidak menjadi terlalu besar, biasanya dinormalisasi dengan akar kuadrat dari dimensi vektor kunci, k_i. Ukuran perhatian produk titik adalah

$$a(q, k_i) = qk_i^T\sqrt{d},$$

di mana k_i^T menunjukkan transpose dari vektor kunci, k_i, dan d adalah panjang vektor k_i. Kita dapat menyederhanakan hasil persamaan ini dengan menggunakan persamaan softmax:

$$a(q, k_i) = \text{softmax}a(q, k\_i) = e^{qk_i^T}\sqrt{d}\{\sum_j e^a(qk_i^T)/\sqrt{d},$$

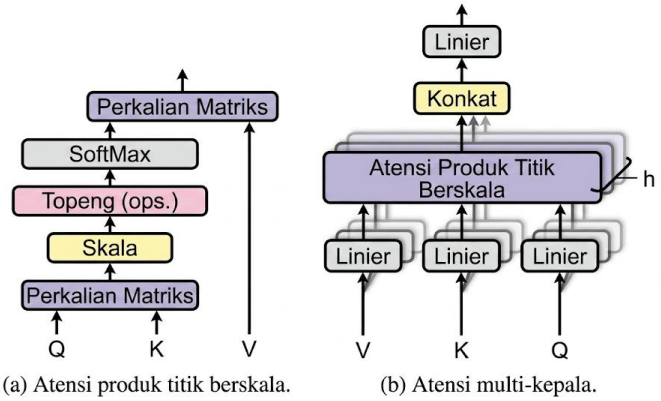
di mana k_i^T dan k_j^T adalah transpos dari k_i dan k_j dan d adalah panjang vektor k_i. Sebagian besar transformer arus, termasuk transformer Vaswani et al. asli, menggunakan produk titik berskala dengan softmax untuk menghitung perhatian. Gambar 17(a) mewakili mekanisme perhatian produk titik berskala. Seperti yang baru saja dicatat, perhatian produk titik berskala dihitung menggunakan representasi matriks Q dan K untuk semua q dan k individual. Dari perspektif matriks, mekanisme perhatian dihitung pada sekumpulan kueri yang digabungkan ke dalam matriks Q. Kunci dan nilai juga digabungkan ke dalam matriks K dan V. Output matriks kemudian adalah

$$\text{Attention}(Q, K, V) = \text{softmax}\frac{(QK^T)}{\sqrt{d_k}}V,$$

di mana T adalah transpose dari K dan d_k adalah dimensi vektor kunci. Alternatif untuk mekanisme perhatian produk titik yang diskalakan adalah menggunakan mekanisme produk titik tanpa faktor penskalaan. Penskalaan digunakan untuk mencegah produk titik menjadi terlalu besar dan membantu menstabilkan proses pembelajaran. Pembagian dengan akar kuadrat dari dimensi vektor kunci tidak memengaruhi distribusi keseluruhan skor perhatian.

Pembelajaran terus berlanjut untuk menentukan mekanisme perhatian yang berkualitas. Bahkan, satu kelompok pembelajaran mengusulkan peningkatan arsitektur transformer dengan bobot perceptron multi-level. Namun, pendekatan yang lebih umum adalah mekanisme perhatian Vaswani et al. asli yang menggunakan pemrosesan perhatian multi-kepala, seperti yang terlihat pada Gambar 1.17(b). Alih-alih menggunakan fungsi perhatian produk titik yang diskalakan dengan kunci, nilai, dan kueri dimensi berbasis model, mereka menemukan bahwa memproyeksikan kueri, kunci, dan nilai secara linier sebanyak h kali dengan proyeksi linier yang dipelajari berbeda akan sangat membantu. Dimensi vektor kunci digunakan untuk kueri dan kunci, dan nilai menggunakan dimensinya sendiri. Dalam proyeksi arsitektur transformer asli mereka, h adalah 8. Pada setiap versi proyeksi kueri, kunci, dan nilai, fungsi perhatian dilakukan secara paralel.

Nilai keluaran perhatian kemudian digabungkan dan diproyeksikan kembali, menghasilkan nilai akhir, seperti yang terlihat pada Gambar 1.16(b). Mekanisme perhatian multi-kepala mendukung gagasan bahwa model dapat memperhatikan subruang representasi yang berbeda pada posisi yang berbeda secara paralel, sebuah fungsi yang tidak mungkin dilakukan dengan satu kepala perhatian pada Gambar 1.17(a). Vaswani dan rekan-rekan. menyajikan beberapa hasil perhatian multi-kepala paralel ini secara grafis.

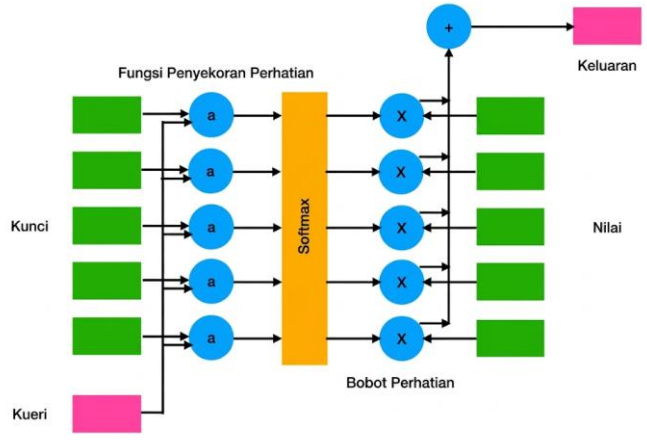


Gambar 1.17. (a) Mekanisme perhatian produk titik berskala standar. (b) menunjukkan mekanisme perhatian produk titik berskala multi-kepala.

Kembali ke deskripsi Gambar 1.13, encoder berisi lapisan self-attention, di mana query, kunci, dan nilai berasal dari output lapisan sebelumnya. Setiap posisi encoder dapat memperhatikan posisi apa pun di lapisan sebelumnya. Demikian pula, lapisan self-attention di

decoder memperhatikan semua posisi hingga dan termasuk posisi saat ini yang ditutupi untuk menegakkan properti auto-regresif. Dalam lapisan perhatian encoder/decoder, query berasal dari lapisan decoder sebelumnya dan kunci serta nilai berasal dari output encoder. Ini memungkinkan setiap posisi di decoder untuk memperhatikan semua posisi dalam urutan encoder. Ini memberi mekanisme perhatian fungsionalitas seperti model sequence-to-sequence pra-transformer sebelumnya yang terlihat pada Gambar 1.12. Agar mekanisme perhatian dapat memanfaatkan urutan dalam urutan, harus ada informasi tentang urutan relatif dan posisi token di setiap urutan. Untuk mencapai hal ini, pengkodean posisi ditambahkan ke penyematan input di bagian bawah tumpukan encoder dan decoder, seperti yang terlihat pada Gambar 1.12. Terakhir, setiap lapisan encoder dan decoder berisi jaringan feed-forward yang terhubung penuh yang diterapkan pada output dari masing-masing mekanisme perhatian. Proses pelatihan untuk tugas penerjemahan transformer Vaswani et al. dijelaskan dalam artikel mereka.

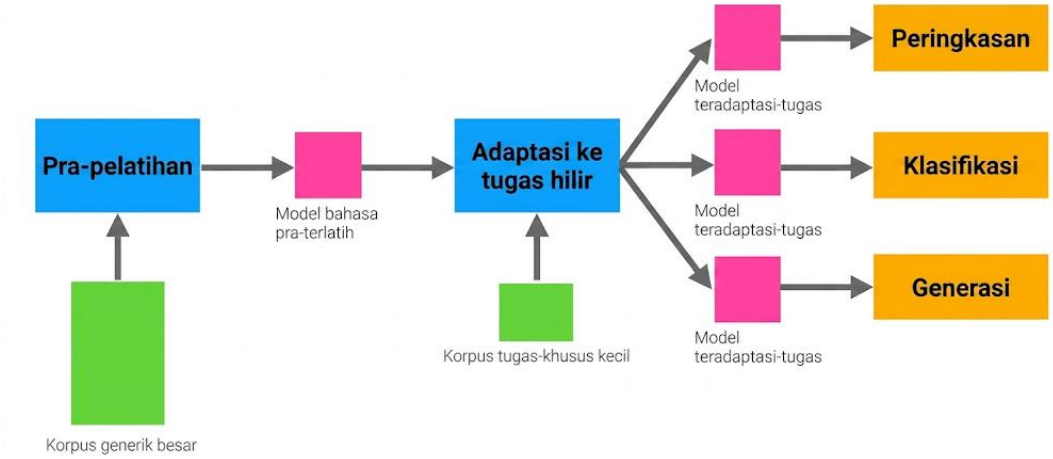
Kami menyimpulkan subbagian ini dengan Gambar 18, representasi dari perhitungan mekanisme perhatian. Untuk deskripsi tingkat tinggi tentang perhatian, bandingkan dengan pengambilan basis data normal. Dalam pengambilan basis data, kueri, q, digunakan untuk mencari kunci, k, yang merupakan penunjuk ke nilai, v, yang ingin kita ambil. Dengan mekanisme perhatian, alih-alih mengembalikan pasangan kunci-nilai tertentu, kita mengembalikan distribusi probabilitas, berdasarkan kueri kita, tentang kesesuaian semua kemungkinan pasangan kunci-nilai. Deskripsi yang lebih formal untuk banyak mekanisme transformer yang dijelaskan di bagian ini dapat ditemukan. Sejak tahun 2018, arsitektur transformer dengan perhatian telah menjadi metodologi dominan yang digunakan dalam membangun model bahasa besar. Praktik penggunaan transformer adalah melatih model skala besar ini terlebih dahulu pada korpus yang sangat besar untuk mengoptimalkan pembelajaran mandiri. Setelah pelatihan awal, model kemudian dapat disempurnakan oleh penggunanya dengan data yang sesuai dengan kebutuhan aplikasi pengguna, seperti yang kita lihat di Bagian 5 dan Gambar 1.19. Ketika menggunakan pendekatan pelatihan awal ini, transformer berbasis perhatian asli disebut sebagai model dasar. Kami mempertimbangkan masalah ini di Bagian 5.



Gambar 1.18 Keluaran dari attention pooling adalah rata-rata tertimbang dari nilai-nilai di mana bobot, a, dihitung menggunakan prosedur perhatian dengan softmax.

1.5 TRANSFORMER DALAM PRAKTIK

Bagian terakhir kami pertama-tama mempertimbangkan bagaimana model berbasis transformer dilatih untuk beroperasi di lingkungan khusus. Kemudian kami mendemonstrasikan penyempurnaan LLM untuk melatihnya untuk domain baru dan kemudian kami menyajikan rekayasa cepat, teknik waktu nyata yang mengkondisikan LLM untuk lebih memfokuskan kinerjanya. Terakhir, kami menjelaskan beberapa area aplikasi penting untuk AI generatif.



Gambar 1.19 Transformer yang telah dilatih sebelumnya dapat mentransfer pengetahuannya ke domain terkait. Domain baru dikondisikan dengan menggunakan dataset spesifik tugas yang lebih kecil.

Pra-pelatihan: Korpus

Model bahasa besar pertama-tama dilatih terlebih dahulu pada korpus data teks dan pada tugas-tugas terkait pemodelan bahasa tertentu. Selama pra-pelatihan, LLM mencoba mempelajari dan memahami pola bahasa umum yang ditemukan dalam hubungan antar kata. Beberapa contoh korpus yang cocok untuk pra-pelatihan LLM:

- Wikipedia Bahasa Inggris — kumpulan artikel dari versi bahasa Inggris Wikipedia, ensiklopedia daring gratis. Ini berisi berbagai topik dan gaya penulisan, menjadikannya sampel representatif dari penggunaan bahasa Inggris. Komponen bahasa Inggris Wikipedia berisi sekitar 2,5 miliar kata.
- *The BookCorpus* — kumpulan besar buku fiksi dan non-fiksi. Ini dibuat dengan mengikis teks buku dari web dan mencakup berbagai genre, dari romansa dan misteri hingga fiksi ilmiah dan sejarah. Buku-buku dalam korpus ini diharuskan memiliki panjang minimal 2000 kata dan ditulis dalam bahasa Inggris. BookCorpus berisi sekitar satu miliar kata.

Penyempurnaan LLM

Pembelajaran transfer adalah teknik yang memanfaatkan pengetahuan yang diperoleh dari satu tugas untuk meningkatkan kinerja pada tugas lain. Transfer learning untuk LLM

melibatkan pengambilan LLM yang telah dilatih sebelumnya dan melakukan fine-tuning untuk tugas "baru" tertentu, seperti klasifikasi teks atau generasi teks. Fine-tuning memperbarui parameter model menggunakan data spesifik tugas baru, seperti pada Gambar 1.19. Fine-tuning terdiri dari empat langkah:

(1) Menentukan model yang akan di-tuning dan parameternya, misalnya laju pembelajaran.
(2) Menggabungkan data pelatihan baru, di mana format dan parameter lainnya bergantung pada model.
(3) Menghitung kerugian, ukuran kesalahan, dan gradien, untuk mengubah model agar meminimalkan kesalahan.
(4) Memperbarui model melalui backpropagation.

Rekayasa Prompt

Rekayasa prompt adalah praktik meminta informasi spesifik dari LLM yang telah dilatih untuk mendapatkan respons yang paling tepat dari model. Ada beberapa pendekatan untuk rekayasa prompt. Permintaan *zero-shot* meminta informasi yang bukan bagian dari pelatihan model; namun, model akan menghasilkan hasil. Teknik ini membuat LLM berguna untuk berbagai tugas. Permintaan *few-shot* adalah strategi di mana model diberi beberapa contoh spesifik tugas sebelum menyajikan kueri sebenarnya. Kueri few-shot memungkinkan model untuk melakukan generalisasi atas kueri. Singkatnya:

> *Zero-Shot Prompting*: Digunakan ketika tugas sudah jelas, tidak memerlukan contoh spesifik.
> *One-Shot Prompting*: Ideal untuk tugas yang membutuhkan format atau konteks tertentu, di mana satu contoh dapat memandu output. Misalnya, berikan deskripsi pekerjaan kepada LLM dan kemudian minta untuk "menulis deskripsi pekerjaan serupa untuk posisi Analis Data".
> *Few-Shot Prompting*: Digunakan untuk tugas kompleks yang membutuhkan banyak contoh yang memberikan konteks lebih luas atau untuk menangani kueri yang lebih bernuansa. Misalnya, setelah memberikan beberapa ulasan produk berlabel, untuk "Memprediksi sentimen ulasan berikut".

Pemberian Petunjuk Rantai Pemikiran: Ini memecah masalah besar menjadi langkah-langkah perantara, memungkinkan model bahasa untuk menangani tugas-tugas kompleks yang tidak dapat diselesaikan dengan teknik pemberian petunjuk standar.

Pemberian petunjuk rantai pemikiran adalah gaya pemberian petunjuk few-shot, di mana petunjuk berisi serangkaian langkah penalaran perantara. Pemberian petunjuk rantai pemikiran mendorong model untuk bernalar sesuai dengan cara petunjuk diajukan, yaitu dalam serangkaian langkah. Anehnya, jawaban dari pemberian petunjuk rantai pemikiran seringkali lebih akurat dan dapat diinterpretasikan daripada jawaban dari petunjuk lainnya. Pemberian petunjuk rantai pemikiran juga mencegah model untuk menghasilkan jawaban cepat dan mudah. Sekarang ada beberapa saran untuk mengatur proses yang baru saja dijelaskan untuk beralih dari model dasar melalui penyempurnaan hingga rekayasa petunjuk.

Proses ini disebut penyelarasan LLM, untuk survei pendekatan. Salah satunya disebut RLHF atau Pembelajaran Penguatan dari Umpan Balik Manusia. Masih ada pertanyaan besar tentang kegunaan pendekatan ini. Selanjutnya, kami mendemonstrasikan penggunaan rekayasa prompt kami sendiri untuk mengubah respons LLM. Untuk contoh kami, kami menggunakan BERT (url 2) dan GPT (url 4). Pengguna memberikan

Prompt: dan LLM membalas dengan ***Output***:. Kami mulai dengan contoh prompt zero-shot:
Prompt: Langit berwarna
Output: biru

LLM membalas dengan kelanjutan string "Langit berwarna", yang mungkin tidak sesuai dengan kebutuhan pengguna. Selanjutnya, kami mencoba untuk mendapatkan hasil yang lebih baik:

Prompt: Lengkapi kalimat: Langit berwarna
Output: sangat indah hari ini.

Karena kita meminta LLM untuk "melengkapi kalimat", hasilnya terlihat berbeda karena LLM mengikuti apa yang diperintahkan. Prompt berisi salah satu elemen berikut dan formatnya bergantung pada tugas:

Instruksi — deskripsi tugas spesifik yang ingin Anda lakukan dengan model.
Konteks — informasi lebih lanjut atau konteks tertentu yang mengarahkan model ke respons.
Data Masukan — masukan atau pertanyaan yang ingin Anda ajukan.
Indikator Keluaran — jenis atau format keluaran, misalnya, puisi.

Kita juga dapat merancang prompt untuk berbagai tugas dengan menggunakan perintah yang menginstruksikan model apa yang ingin kita capai. Prompt ini meliputi: "Tulis", "Klasifikasi", "Ringkas", "Terjemahkan", "Urutkan", dll.

Prompt: Instruksi: Terjemahkan teks ke bahasa Spanyol: Teks: "halo!"
Keluaran: Hola!

Kita dapat menambahkan batasan kontekstual pada perintah untuk memberikan informasi kepada model atau untuk membatasi batasan respons terhadap perintah tersebut. Konteks dapat berupa instruksi yang menentukan bagaimana model harus berperilaku atau informasi yang digunakan atau dirujuk oleh model untuk menghasilkan respons. Misalnya:

Perintah: Kelereng:
Warna: merah Jumlah: 12
Warna: biru Jumlah: 28
Warna: kuning Jumlah: 15

Warna: hijau Jumlah: 17
Berapa banyak kelereng hijau yang ada?
Output: Ada 17 kelereng hijau.

Kita juga dapat menambahkan contoh untuk memperkaya konteks.

Perintah: Klasifikasikan yang berikut ini.
Pilihan:
- anggur merah
- anggur putih

Teks: Chardonnay
Jawabannya adalah: anggur putih
Teks: Cabernet
Jawabannya adalah: anggur merah
Teks: Moscato
Jawabannya adalah: anggur putih
Teks: Riesling
Jawabannya adalah:
Output: anggur putih.

Untuk menunjukkan kegunaan pemberian perintah few-shot, pertimbangkan *analisis sentimen*: Kita mengambil paragraf dengan berbagai pendapat dan memberi label pada paragraf tersebut dengan klasifikasi sentimen. Kemudian kita mengkondisikan model untuk menerima paragraf sebagai input dan menghasilkan klasifikasi sebagai output. Sebelum LLM berbasis transformer, jaringan saraf berulang tidak terlalu adaptif. Menambahkan klasifikasi baru untuk dipertimbangkan atau meminta model untuk tidak mengklasifikasikan tetapi meringkas paragraf membutuhkan modifikasi dan pelatihan ulang model.

Namun, model bahasa transformer yang besar tidak perlu dilatih ulang. Kita dapat meminta model untuk mengklasifikasikan paragraf atau meringkasnya ketika kita tahu cara meminta dengan benar. Model tidak dapat mengklasifikasikan paragraf ke dalam kategori A atau B ketika spesifikasi untuk A dan B tidak jelas. Seperti yang akan kita lihat selanjutnya, LLM dapat mengklasifikasikan sentimen "positif" atau "negatif" setelah model diberi tahu apa itu "positif" dan "negatif". Selama pelatihan, model mempelajari kata-kata ini dan memperoleh kemampuan untuk mengikuti instruksi sederhana.

Instruksi: Klasifikasikan teks ke dalam positif, netral, atau negatif: Teks: Pertunjukan musik itu menyenangkan.
Klasifikasi:
Output: Teks "Pertunjukan musik itu menyenangkan" dapat diklasifikasikan sebagai positif.

Responsnya adalah "positif". Ini benar dan ringkas. Model juga akan dapat mengklasifikasikan "luar biasa" sebagai positif. Mengidentifikasi sentimen dihasilkan dari instruksi awal: Klasifikasikan teks ke dalam positif, netral, atau negatif.

Instruksi: Teks: Ruangan itu sangat panas.
Klasifikasi: Negatif
Teks: Ceramah itu sangat intens.
Klasifikasi:

Output: Teks "Ceramah itu sangat intens" dapat diklasifikasikan sebagai netral. Ini menggambarkan intensitas ceramah tanpa secara inheren menyampaikan sentimen positif atau negatif.

***Instruksi*:** Teks: Hari ini cuacanya fantastis
Klasifikasi: Positif
Teks: Perabotannya kecil.
Klasifikasi: Negatif
Teks: Saya tidak suka sikap Anda
Klasifikasi: Negatif
Teks: Pemilihan bidikan itu mengerikan
Klasifikasi:

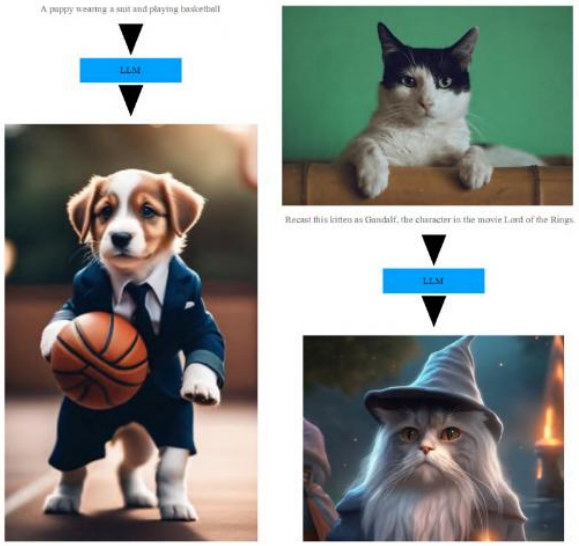
Output: Teks "Pemilihan bidikan itu mengerikan" dapat diklasifikasikan sebagai negatif, karena menyatakan ketidaksetujuan atau kritik yang kuat terhadap pemilihan bidikan tersebut.

Selanjutnya, penalaran rantai pikiran memungkinkan model untuk menguraikan masalah kompleks menjadi langkah-langkah perantara yang diselesaikan secara individual. Selain itu, sifat berbasis bahasa dari rantai pikiran membuatnya dapat diterapkan pada sebagian besar tugas yang dapat diselesaikan orang menggunakan bahasa.

Prompt: Manakah cara yang lebih cepat untuk pulang?
Pilihan 1: Naik bus 10 menit, lalu bus 40 menit, dan akhirnya kereta 10 menit.
Pilihan 2: Naik kereta 90 menit, lalu bersepeda 45 menit, dan akhirnya bus 10 menit.
Pilihan 1 akan memakan waktu 10 + 40 + 10 = 60 menit. Pilihan 2 akan memakan waktu 90 + 45 + 10 = 145 menit.
Karena Pilihan 1 memakan waktu 60 menit dan Opsi 2 membutuhkan waktu 145 menit, Opsi 1 lebih cepat.
Mana cara yang lebih cepat untuk pergi bekerja?
Opsi 1: Naik bus selama 1000 menit, kemudian kereta api setengah jam, dan bersepeda selama 10 menit.
Opsi 2: Naik bus selama 800 menit, kemudian kereta api satu jam, dan terakhir bersepeda selama 30 menit.

Output: Opsi 1 akan membutuhkan waktu 1000 + 30 + 10 = 1040 menit.
Opsi 2 akan membutuhkan waktu 800 + 60 + 30 = 890 menit.
Karena Opsi 2 membutuhkan waktu 890 menit dan Opsi 1 membutuhkan waktu 1040 menit, Opsi 2 lebih cepat.

Pada Gambar 1.20, kita meminta LLM untuk membuat dua gambar, yang pertama dari deskripsi kata dan yang kedua dari gambar ditambah deskripsi kata.



Gambar 1.20 LLM menciptakan gambar anak anjing yang mengenakan jas dan bermain basket (dengan 5 kaki?). Gambar kedua mengubah anak kucing menjadi klon Gandalf. Gambar diadaptasi dari.

Contoh aplikasi AI generatif

Dengan menggunakan konsep dasar yang mendukung pelatihan LLM dan rekayasa cepat, beberapa aplikasi menarik telah muncul. Model AI generatif digunakan untuk menulis novel, lulus ujian pengacara, membuat materi kampanye pemasaran, secara otomatis meregenerasi laporan keuangan, dan untuk banyak tugas menarik lainnya. LLM memiliki aplikasi di luar domain pembuatan teks. LLM dapat digunakan untuk menghasilkan gambar dari teks, atau deskripsi teks dari gambar, atau, dengan instruksi, untuk mengubah gambar.

Konversi Gambar ke Gambar

Konversi gambar mencakup transformasi elemen eksternal gambar, seperti warna, media, atau bentuknya, sambil mempertahankan elemen konstituennya, seperti pada Gambar 20. Salah satu contohnya adalah mengubah gambar siang hari menjadi gambar malam hari. Konversi jenis ini juga digunakan untuk memanipulasi atribut gambar, seperti wajah, mewarnainya, atau mengubah gayanya. Contoh lain adalah mengambil gambar dengan gaya seorang pelukis dan mengubahnya dengan gaya pelukis lain. Konversi gambar juga telah digunakan dalam aplikasi pengeditan foto tingkat lanjut, misalnya untuk menambah atau

menghapus detail, figur, atau objek dari gambar. Semua tugas ini dilakukan menggunakan petunjuk yang dirancang dengan cermat.

Teks ke Gambar dan Gambar ke Teks

Model AI generatif juga dilatih untuk menghasilkan gambar dari deskripsi teks di bawah kelas model khusus yang disebut difusi. Model difusi adalah kelas model generatif yang mensimulasikan proses pembangkitan data dengan mengubah distribusi awal yang sederhana menjadi distribusi data kompleks yang diinginkan melalui serangkaian operasi yang dapat dibalik. Model ini dapat digunakan dalam aplikasi termasuk menjawab pertanyaan tentang gambar, membuat deskripsi gambar, dan mengedit gambar.

Teks ke Video

Menariknya, transformer tidak hanya dapat menghasilkan teks ke gambar, seperti yang kita lihat pada Gambar 1.20, tetapi pembelajaran saat ini sedang mengeksplorasi kemungkinan mengubah teks menjadi video. Misalnya, ``Tunjukkan padaku seorang wanita dengan parka merah sedang bermain ski di lereng resor" atau ``Seorang pria dan wanita dengan pakaian hitam sedang melakukan ice dancing". Banyak pertanyaan penting yang masih belum terjawab, termasuk bagaimana cara terbaik untuk melatih transformer dengan materi video, lihat juga (url 15).

Produksi Musik

Dengan menggunakan jaringan saraf berulang dan autoencoder variasional, AI generatif telah dilatih untuk secara otomatis membuat musik dari deskripsi teks. Model dapat mempelajari motif melodi, progresi akord, serta elemen ritmis. Aplikasinya meliputi sintesis musik otomatis dari berbagai genre, penyesuaian komposisi, pengajaran teori musik, dan memicu kreativitas pada pembuat musik.

Generasi Kode

AI generatif dapat digunakan untuk membuat kode komputer berdasarkan perintah. Kode ini dapat dihasilkan di sebagian besar bahasa pemrograman hanya dengan menggunakan deskripsi bahasa alami dari masalah tersebut. Aplikasi mendukung saran penyelesaian kode yang lebih akurat, menghasilkan uji unit untuk suatu fungsi, debugging otomatis, menghasilkan dokumentasi, mengajarkan pemrograman, dan menjawab pertanyaan tentang bagian-bagian kode.

Desain dan Generasi Protein

Model AI terlatih digunakan untuk menghasilkan urutan protein baru. Pelatihan terdiri dari beberapa urutan asam amino dari protein yang berbeda dan kemudian penyempurnaan dengan subset urutan yang lebih kecil yang memiliki deskripsi kontekstual. Dengan menggunakan petunjuk bahasa untuk properti yang diinginkan dari protein yang dibutuhkan, model secara otomatis menghasilkan beberapa protein yang urutan asam aminonya memenuhi properti tersebut. Teknologi ini telah digunakan untuk mempercepat proses desain obat untuk obat-obatan baru.

Ekstensi Lanjutan dari Pemecah Masalah DeepMind Google

Kelompok riset DeepMind Google merancang keluarga program permainan Alpha. Program Alpha Zero mereka memainkan go, catur, dan shogi semuanya pada tingkat kelas

dunia. Ekstensi terbaru, AlphaGeometry, menggabungkan AI simbolik dengan teknologi model bahasa pembelajaran mendalam untuk menghasilkan bukti dalam geometri Euclidean. AI simbolik, sejalan dengan program pembuktian teorema matematika tradisional, menyarankan prosedur pembuktian yang mungkin. Solusi parsial ini kemudian diteruskan ke model pembelajaran mendalam untuk mengeksplorasi metode simbolik dan berbasis konstruksi untuk menyelesaikan pembuktian. Program yang dihasilkan melampaui kemampuan pembuktian geometri komputer sebelumnya dan mendekati kinerja para pesaing Olimpiade Matematika Internasional. Terakhir, Tabel 1.4 menyajikan model bahasa besar yang paling banyak digunakan saat ini, beserta beberapa parameter desainnya. Dalam tabel tersebut, "token" mengacu pada informasi yang digunakan untuk melatih model, suku kata, kata, komponen gambar, dll. Memiliki lebih banyak parameter atau token pelatihan tidak selalu mendukung model yang lebih baik, karena desain kode dan efisiensi komputasi selalu tetap penting.

1.6 KESENJANGAN ANTARA INOVASI DAN AKUNTABILITAS DALAM AI

Kami telah menyajikan gambaran umum tentang evolusi pembelajaran dan praktik jaringan saraf, dimulai pada tahun 1940-an dengan model pemrosesan kortikal manusia dan diakhiri dengan model bahasa besar berbasis transformer. Kisah lengkap evolusi ini membutuhkan analisis algoritma dan praktik rekayasa yang jauh lebih mendalam daripada yang telah kami sajikan di sini. Meskipun aplikasi teknologi ini saat ini sangat mengesankan, seperti yang kita lihat di Bagian 5, dan menawarkan peluang baru bagi praktisi AI, masih ada masalah penting yang harus ditangani masyarakat.

Pertama, terdapat kurangnya integritas matematis dan dukungan untuk banyak praktik rekayasa AI modern. Bahwa para insinyur menciptakan program "karena tampaknya berhasil" bukanlah pembenaran yang cukup untuk alat perangkat lunak yang mulai memainkan peran penting dalam kehidupan modern. Yang menambah masalah ini adalah fakta bahwa banyak pencipta LLM kita saat ini merahasiakan praktik rekayasa mereka sebagai "rahasia perusahaan". Melindungi keuntungan untuk investasi perusahaan itu penting, tetapi akuntabilitas untuk penggunaan algoritma dan keputusan rekayasa sangat penting untuk penggunaan teknologi ini secara bertanggung jawab oleh masyarakat. Langkah ke arah ini adalah dengan meningkatkan kolaborasi universitas, publikasi yang ditinjau oleh rekan sejawat, dan analisis terbuka terhadap praktik AI generasi saat ini. Dan terkait dengan hal ini, tentu saja, adalah pendidikan yang lebih baik bagi masyarakat tentang janji dan masalah dari generasi pemecah masalah AI saat ini.

Langkah pertama yang penting di sini adalah mengakui bahwa LLM mencerminkan data yang digunakan untuk melatihnya. Jika data tersebut rasis dan/atau seksis, maka produknya juga akan demikian. Jika model dilatih pada data dari tanggal dan lokasi tertentu, hasilnya akan mencerminkan hal ini. Penyesuaian dan rekayasa cepat LLM untuk mengekstrak hasil yang bermanfaat lebih merupakan seni gelap daripada sains, meskipun beberapa kelompok pembelajaran sedang menangani hal ini. Model bahasa besar relatif bebas pengetahuan. Mereka tidak tahu apa yang mereka ketahui dan, lebih buruk lagi, mereka tidak tahu apa yang

tidak mereka ketahui. Tetapi mereka akan selalu menawarkan respons kepada pengguna. Pertimbangkan kembali anjing berkaki lima pada Gambar 1.20. "Pengetahuan" mereka adalah bagaimana kata/token diasosiasikan dalam korpus besar, dengan jaringan perhatian membantu membangun asosiasi yang lebih kompleks. Model-model ini, dalam pengertian manusia, tidak "mengetahui" apa pun.

Kritik lebih lanjut, yang dijelaskan oleh tidak adanya pengetahuan dalam sistem, adalah kurangnya transparansi dan penjelasan dari LLM dan produk-produknya. Ketika sebuah LLM menghasilkan suatu produk, hampir tidak mungkin baginya untuk memberikan justifikasi atas hasil tersebut. Untuk sebagian besar keputusan penting masyarakat, justifikasi dan penjelasan diperlukan. Mengapa saya disetujui untuk pinjaman bank tersebut dengan suku bunga tertentu? Mengapa Anda berpikir tumor ini ganas? Mengapa pembelian saham ini direkomendasikan? Pengambilan keputusan yang bertanggung jawab secara manusiawi membutuhkan transparansi dan penjelasan. Pemerintah Amerika Serikat (url 20, url 21, url 22) dan Uni Eropa masing-masing telah mengusulkan pedoman untuk penggunaan AI (url 23) seperti halnya perusahaan-perusahaan besar termasuk IBM (url 19) dan Microsoft (url 18). Masyarakat profesional termasuk IEEE serta buku teks AI modern yang berisi bab-bab tentang penggunaan etis teknologi AI modern. Kami mencatat dalam pendahuluan kami bahwa banyak pelopor ilmu saraf jaringan saraf, termasuk McColloch dan Pitts, Hebb, dan Rosenblatt, merasa bahwa ciptaan mereka meniru aspek pemrosesan neuron manusia. Visi itu bukan lagi bagian dari prospek generasi LLM saat ini.

BAB 2
MASALAH, SOLUSI, DAN KOMPUTASI SEMANTIK

2.1 INTEGRASI MASALAH DAN PENAWARAN MELALUI KOMPUTASI SEMANTIK

Komputasi Semantik memperluas Web Semantik baik dalam cakupan maupun kedalaman. Ia menjembatani, dan mengintegrasikan, beberapa teknologi komputasi ke dalam tema yang lengkap dan terpadu. Artikel ini membahas esensi Komputasi Semantik dengan deskripsi SemanticServices.Net, paradigma baru yang memungkinkan pencarian berbasis masalah yang dapat menawarkan cerita baru bagi Internet. Dunia komersial sebagian besar berkaitan dengan permintaan dan penawaran. Dalam kebanyakan kasus, permintaan memicu penawaran, dan dalam beberapa kasus, penawaran menciptakan permintaan. Konsep yang lebih umum untuk kebutuhan mungkin berupa masalah, dan untuk penawaran mungkin berupa solusi. Internet telah menyediakan infrastruktur global untuk menghubungkan masalah dengan solusi. Misalnya, eBay telah melakukan pekerjaan yang hebat dalam lelang. Mesin pencari berbasis kata kunci seperti Google dapat dianggap sebagai pemecah masalah khusus yang memecahkan masalah: Temukan dokumen (Web) yang berisi kata kunci yang diberikan oleh pengguna. Sistem Tanya Jawab (Q&A) dapat dianggap sebagai pemecah masalah khusus lainnya yang memecahkan masalah: Temukan jawaban untuk pertanyaan (berdasarkan dokumen yang dikumpulkan oleh sistem).

Baik mesin pencari berbasis kata kunci maupun Sistem tanya jawab telah melakukan pekerjaan yang sangat baik untuk masalah yang mereka coba selesaikan. Tetapi dari sudut pandang pemecahan masalah, sistem tersebut masih jauh dari memadai. Dapat dengan mudah dilihat bahwa tidak setiap masalah termasuk dalam dua kategori umum yang telah kita bahas. Setiap ilmuwan komputer dapat dengan mudah membuat daftar berikut:

(1) Masalah Komputasi dan Masalah Matematika lainnya. Memecahkan masalah seperti itu membutuhkan komputasi. Beberapa upaya awal telah dilakukan, tetapi masih banyak yang perlu dilakukan di bidang ini.
(2) Masalah Pencarian Basis Data seperti Temukan supermarket yang menjual apel dengan harga kurang dari Rp 20.000 per pon. Deep web biasanya bekerja sendiri dan tidak terhubung (bukan karena tidak bisa, tetapi mungkin karena mereka tidak ingin terhubung).
(3) Masalah Sintesis seperti Buat program yang mengambil sekumpulan angka dan mengembalikannya dalam urutan menaik. Sintesis otomatis secara umum sulit dan tetap menjadi tujuan yang harus dicapai.
(4) Masalah Penalaran seperti Apa yang dapat diturunkan dari himpunan ini
(5) Masalah Analisis Data seperti Apa pola umum yang ditunjukkan dalam kumpulan gambar ini? Ada banyak pendekatan yang dapat kita ambil untuk menyelesaikan masalah ini; tetapi ini bukan masalah yang ditangani oleh mesin pencari atau sistem tanya jawab.

(6) Masalah "Pribadi" seperti Saya mengenal orang A dan orang B tetapi mereka tidak saling menyukai, bagaimana saya dapat membuat mereka bekerja? Ini mungkin bukan masalah ilmiah dan solusinya sangat bergantung pada pengalaman, pertimbangan sosial, dll.

Inti utama kami bukanlah mengklasifikasikan semua masalah. Yang sebenarnya menarik bagi kami adalah mencocokkan masalah dengan solusi. Internet memang menyediakan infrastruktur untuk menghubungkan masalah dan solusi, tetapi kita mungkin belum sepenuhnya memanfaatkan infrastruktur ini. Sejauh ini, infrastruktur ini bermanfaat untuk perdagangan. Jika kita dapat memperluas konsep perdagangan dari barang ke masalah dan pembeli ke solusi, kita mungkin akan memiliki kisah baru untuk Internet.

2.2 MEKANISME INTEGRASI DALAM EKOSISTEM PERTUKARAN SOLUSI

Beberapa faktor dapat dianggap penting untuk keberhasilan infrastruktur yang mendukung "pertukaran" antara masalah dan solusi:

(1) Orang (Penyedia) harus bersedia menyediakan solusi. Secara khusus, mereka harus mendapat manfaat dari pertukaran tersebut.

(2) Orang (Konsumen) harus bersedia menyampaikan masalah. Jika solusi tersedia dengan mudah, konsumen dapat memperoleh manfaat dari solusi tersebut. Masalah tanpa solusi dapat berkontribusi pada pengembangan solusi untuk masalah tersebut di kemudian hari.

(3) Infrastruktur harus mengakomodasi penyedia dan konsumen dari semua ukuran sehingga siapa pun yang memiliki solusi untuk suatu masalah bersedia menjadi penyedia, dan siapa pun yang memiliki masalah untuk dipecahkan kemungkinan akan menggunakan infrastruktur tersebut.

(4) Penyedia biasanya ingin melindungi "pengetahuan" tentang suatu solusi jika memungkinkan.

(5) Suatu solusi tidak harus berupa alat; dalam banyak kasus, solusi hanya melibatkan konten (seperti teks, gambar, video, proses, perangkat keras, perangkat lunak, dll.) dan pengetahuan.

(6) Suatu mekanisme mungkin diperlukan untuk memvalidasi solusi dan untuk berbagi umpan balik dari konsumen tentang solusi tersebut.

Secara teknis, faktor keempat menunjukkan bahwa infrastruktur menggunakan arsitektur layanan — satu atau lebih solusi dikemas sebagai layanan. Keuntungan dari layanan adalah bahwa internalnya dapat disembunyikan dari konsumen. Selain itu, untuk mencapai tujuan kita, tiga komponen diperlukan:

(1) Deskripsi Masalah

(2) Deskripsi Solusi

(3) Pencocokan Masalah dengan Solusi

Tampaknya kata kunci tidak cukup untuk menggambarkan masalah dan solusi jika kita ingin mengakomodasi sejumlah besar masalah dan solusi pada granularitas yang halus, seperti yang disarankan oleh faktor 1, 2, 3, dan 5.

2.3 MENDESKRIPSIKAN SOLUSI DALAM SCDL

Yang penting untuk suatu solusi (layanan) bukanlah bagaimana cara kerjanya; melainkan apa yang dapat dilakukannya. Kita menyebutnya kemampuan solusi (layanan). *Semantic Capability Description Language* (SCDL) adalah bahasa deskripsi mirip SQL yang dapat digunakan untuk mendeskripsikan kemampuan suatu solusi (layanan), dengan tujuan untuk mendukung komposisi layanan otomatis. Sintaks SCDL untuk suatu layanan mirip dengan SQL, seperti yang dinyatakan dalam bentuk generik berikut:

SELECT outputs $(O_1; \ldots; O_m)$, aggregated $-$ outputs $(f_1(A_1); \ldots; f_d(A_d))$
FROM inputs $(I_1; \ldots; I_m)$, variables $(R_1; \ldots; R_n)$, other variables $(S_1; \ldots; S_k)$
WHERE p (inputs, outputs, other variables)
GROUP BY $(H_1; \ldots; H_j)$

di mana $O_1; \ldots; O_m$ adalah objek output. $f_1(A_1); \ldots; f_d(A_d)$ adalah fungsi agregasi yang mungkin, $I_1; \ldots; I_m$ adalah objek input, $R_1; \ldots; R_n$ adalah beberapa variabel rentang, $S_1; \ldots; S_k$ adalah himpunan yang dapat diturunkan dari input dan variabel rentang, $H_1; \ldots; H_j$ adalah variabel untuk mengelompokkan objek output, dan p (input, output, variabel lain) adalah rumus yang menggambarkan hubungan antara input, output, dan variabel. SCDL memungkinkan variabel untuk diketik, dan memungkinkan fungsi untuk dimasukkan sebagai kondisi dalam klausa WHERE. Selain itu, SCDL memungkinkan "variabel eksponensial", di mana domain variabel eksponensial dapat berupa himpunan semua himpunan bagian dari himpunan yang ada, dan memungkinkan variasi variabel eksponensial (kita akan melihat beberapa contoh nanti). Ekspresi aljabar yang sesuai adalah sebagai berikut:

$$WS(I_1, \ldots, I_m; O_1, \ldots, O_m) =_{(H_1,\ldots,H_j)} G_{f_1\Lambda_1,\ldots,f_d\Lambda_d)} \times \Pi\left(\sigma_p R_1 \times \ldots \times R_n \times S_1 \times \ldots \times S_k\right))$$

Sebagai contoh, jika seseorang memiliki alat yang dapat mendeteksi gugusan blob dalam sebuah gambar yang terstruktur seperti satelit atau bintang, ia dapat mengemas alat tersebut sebagai layanan dengan deskripsi SCDL berikut:

$$[S1 - SCDL]$$
$$SELECT\ i$$
$$FROM\ INPUT\ setof - image\ datain,$$
$$setof - string\ s\ =\ ['satellite', 'star'$$
$$image{:}\ dataset\ i,$$
$$setof - blob{:}\ 2^{i}.^{blobs()}\ c$$
$$INPUTstring{:}\ s\ t$$
$$WHERE\ i.contain(c) and\ c.islike(t)$$

Dalam contoh di atas, `*image:dataset i*` menyatakan bahwa `i` adalah objek bertipe `*image*` yang domainnya adalah himpunan `*dataset*`, `*setof-string s*=[`*satellite',`star'* ] s` menyatakan bahwa `s` adalah himpunan string `[`*satellite',`star'*]`, `*string:s t*` menyatakan bahwa `t` adalah string yang domainnya adalah `s`, `*setof-blob:2i.blobs()c*` menyatakan bahwa `c` adalah gugus blob dalam `i`, kondisi `i.contains(c)` bernilai benar jika `c` berada dalam `*image*` `i`, dan kondisi `*c.islike*(s)` bernilai benar jika `c` secara struktural menyerupai satelit atau bintang. Input dari layanan ini adalah dataset gambar, dan outputnya adalah subset gambar yang berisi satu atau lebih gugus blob yang terstruktur seperti satelit atau bintang. Jika alat ini tidak hanya dapat mendeteksi gugus blob yang terstruktur seperti satelit atau bintang tetapi juga dapat memberi anotasi pada gugus tersebut, maka alat ini dapat dikemas sebagai layanan dengan deskripsi SCDL berikut:

$$
\begin{aligned}
&[S2 - SCDL]\\
&SELECT\ annotate(c)\\
&FROM\ INPUT\}\ setof - image\ datain,\\
&\qquad setof - string\ s\ =\ ['satellite', 'star'\\
&\qquad image\colon datain\ i,\\
&\qquad setof - blob\colon 2^{i}.^{blobs()}\ c\\
&\qquad INPUT\ string\colon s\ t\\
&WHERE\ i.contain(c)\ and\ c.islike(t)
\end{aligned}
$$

Sebagai contoh lain, jika seseorang memiliki alat yang dapat mendeteksi gugusan blob dalam sebuah gambar yang tidak tumpang tindih, ia dapat mengemas alat tersebut sebagai layanan dengan deskripsi SCDL berikut:

$$
\begin{aligned}
&[S3 - SCDL]\\
&SELECT\ i\\
&FROM\ INPUT\ setof - image\ datain,\\
&\qquad image\colon dataset\ i,\\
&\qquad setof - blob\colon\ 2^{i}.^{blobs()}\ c\\
&\qquad setof - blob\colon\ 2^{i}.^{blobs()}\ c\\
&WHERE\ i.contain(c)\ and\ i.contain(s) andnot\ overlap(c, s)
\end{aligned}
$$

Suatu layanan tidak harus berupa alat; layanan juga bisa berupa konten. Pertimbangkan, misalnya, kasus di mana seseorang memiliki koleksi klip video tentang berbagai acara yang melibatkan Presiden Obama. Ia dapat mengemas video-video tersebut sebagai layanan dengan deskripsi SCDL berikut:

$$
\begin{aligned}
&SELECT\ v\\
&FROM\ setof - image\ dataset\ =\ 'my - dataset',\\
&\quad video\colon dataset\ v\\
&event\ e,\\
&\quad INPUT\ string\colon ['sport', 'political', 'academic']\ activity
\end{aligned}
$$

$$WHERE\ v.contain-event(e) and\ agent(e,\ 'President\ Obama') and\ category(e, activity)$$

Selain alat dan konten, layanan juga bisa berupa hal lain. Misalnya, seseorang mungkin tahu cara melatih anjing dengan baik. Ia dapat mempublikasikan hal ini sebagai layanan dengan deskripsi SCDL berikut:

$$SELECT g\ =\ trained(k)$$
$$FROM\ INPUT dog\ k, VAR\ dog\ g$$
$$WHERE\ *$$

Pada contoh di atas, k adalah anjing yang akan dilatih, g adalah anjing setelah pelatihan? `*' adalah karakter pengganti yang menunjukkan ``kondisi sembarang pada k dan g." Perhatikan bahwa suatu layanan tidak harus diberikan secara online; SCDL hanyalah bahasa spesifikasi. Terakhir, SCDL dapat diperluas untuk menyertakan meta-variabel untuk menggambarkan layanan yang cakupannya lebih luas. Misalnya, dengan empat kelas yaitu akun, pelanggan, cabang, dan depositor, ekspresi SCDL berikut menentukan bahwa ekspresi tersebut dapat menerima kueri SQDL apa pun yang memilih objek dari kelas-kelas ini dengan kombinasi perbandingan relasional apa pun antara atributnya, tetapi jumlah kondisinya tidak boleh melebihi 4.

```
[S4 − SCDL]
SELECT META select
FROM META from
WHERE META where
META IF member(t, select)
            THEN member(t. path, path(account) union
              path(branch) union path(customer) union
              path(depositor))
META IF member(t, from)
     THEN member (t. domain, [account, branch,
     customer, depositor])
META IF member(t, where) AND member(s, t. arguments)
                AND isa(s, variable)
     THEN member(s. path, path(account) union path(branch)
union path (customer)
          union path(depositor))
META IF member(t, where)
            THEN member(t. predicate, [<, >, =, ! =, >=, <=])
META cardinality(where)  <=  4
```

Pada contoh di atas, select; from dan where adalah meta-variabel yang menurut definisi adalah himpunan. Terdapat lima batasan mengenai meta-variabel. Batasan pertama pada dasarnya menyatakan bahwa meta-variabel select dapat berupa himpunan atribut yang sah;

batasan kedua menyatakan bahwa meta-variabel from dapat berupa himpunan variabel yang dideklarasikan di bawah empat kelas objek; batasan ketiga dan keempat menyatakan bahwa meta-variabel where dapat berupa himpunan kondisi antara dua atribut atau antara atribut dan konstanta, selama atribut tersebut sah dan perbandingannya adalah perbandingan relasional; akhirnya batasan kelima menyatakan bahwa jumlah kondisi tidak boleh melebihi empat.

Contoh lebih lanjut yang menunjukkan bagaimana SCDL dapat digunakan untuk mendeskripsikan kemampuan layanan untuk setiap jenis masalah yang dibahas di Bagian 1 dapat ditemukan. Perhatikan bahwa meskipun ekspresi SCDL dapat dieksekusi, dalam praktiknya seringkali tidak realistis untuk melakukannya. Bahasa ini digunakan untuk mendeskripsikan kemampuan solusi saja. Dengan membandingkan kemampuan layanan yang dinyatakan dalam SCDL dan deskripsi masalah, kecocokan dapat ditentukan.

2.4 MENDESKRIPSIKAN MASALAH DALAM SQDL

SQDL mirip dengan SCDL, kecuali bahwa semua variabel input diinstansiasi ke konstanta. Sebuah kueri dalam SQDL disajikan sebagai: *SELECT objek, atribut objek dan/atau fungsi FROM kelas objek* [WHERE *kondisi*] Sekarang, masalah ``*Tampilkan semua klaster blob dari gambar dalam dataset `cmd-232' yang terstruktur seperti satelit atau bintang*" dapat dideskripsikan dalam SQDL sebagai:

$$
\begin{aligned}
&[\text{SQDL1}]\\
&SELECT\ i\\
&FROM\ image\text{:}\ 'cmd-232'\\
&\quad im\ setof-blob\text{:}\ 2^{im.blobs()}sb\\
&\quad string\text{:}\ ['satellite','star']\ t\\
&WHERE\ im.contain(sb)\ and\ sb.islike(t)
\end{aligned}
$$

Sebagai contoh lain, masalah ``*Tampilkan semua gugus blob dari gambar dalam dataset `cmd-232' yang terstruktur seperti satelit atau bintang dan tidak tumpang tindih dengan struktur blob yang diberikan*" dapat dijelaskan dalam SQDL sebagai:

$$
\begin{aligned}
&[SQDL2]\\
&SELECT\ i\\
&FROM\ image\text{:}\ 'cmd-232' im,\\
&\quad setof-blob\text{:}\ 2^{im.blobs()}sb\\
&\quad setof-blob\text{:}\ 2^{im.blobs()}sc\\
&\quad string\text{:}\ ['satellite','star'\ t\\
&WHERE\ im.contain(sb) and\ sb.islike(t) and\ not\ overlap(sb,sc)
\end{aligned}
$$

Dalam uraian di atas, SQDL1 dapat diselesaikan hanya dengan S1 di Bagian 4. Namun SQDL2 harus diselesaikan dengan menggabungkan S1 dan S3.

2.5 SNL SEBAGAI ANTARMUKA ORA BAGI PENGGUNA

Baik SCDL maupun SQDL tampak sulit dipahami oleh pengguna biasa. Diperlukan bahasa yang lebih sederhana agar konsumen dan penyedia dapat menyusun kalimat mereka. Bahasa Alami Terstruktur (SNL) adalah subset dari bahasa alami yang kekuatan ekspresifnya dijelaskan oleh Aljabar Relasional Objek (ORA). Dapat ditunjukkan bahwa setiap kueri yang dinyatakan dalam SQDL dapat dinyatakan dalam ORA, tetapi sebaliknya tidak benar. Ini mungkin merupakan kompromi yang diperlukan yang harus kita lakukan untuk mendorong lebih banyak konsumen (dan penyedia) untuk bergabung dengan infrastruktur.

Dunia ORA terdiri dari objek yang bertipe. Terkait dengan objek, kita dapat mendefinisikan serangkaian predikat dan operasi. Predikat dan operasi ini (secara kolektif disebut metode) dapat digunakan dalam bahasa kueri yang unik untuk memanipulasi serta mengambil data. Hal ini memungkinkan banyak konsep abstrak atau metode yang bergantung pada domain yang tidak dapat diekspresikan dalam model relasional untuk dimasukkan sebagai bagian dari kueri. Detail ORA dapat ditemukan.

Sebuah kalimat dalam SNL diekspresikan dalam bentuk berikut:
Kata kerja kata benda di mana kata sifat

Kalimat SNL dapat disusun oleh pengguna awam berdasarkan pilihan hierarkis ganda sederhana tanpa mengetahui konsep tingkat rendah seperti "gabung" dan "seleksi". Misalnya, anggaplah seseorang ingin mengidentifikasi jumlah pasien dengan diagnosis klinis penyakit Alzheimer yang menderita diabetes dan telah dihasilkan untuk penanda genetik tertentu (misalnya ApoE) hasil genotipe yang tersedia. Kalimat tersebut akan terdiri dari kata benda (*pasien*), tiga kata sifat/predikat ("dengan diagnosis klinis AD", "menderita diabetes", "memiliki hasil genotipe ApoE yang tersedia") dan kata kerja ("temukan"). Perhatikan SQDL1 yang dibahas di Bagian 4. Ini dapat dijelaskan dalam SNL sebagai:

[SNL1] *Tampilkan (VERB) gumpalan gambar di `cmd-232' (NOUN) yang memiliki gumpalan yang terstruktur seperti satelit atau bintang (ADJECTIVE)*

Dan ekspresi SNL untuk SQDL2 adalah:

[SNL2] *Tampilkan (VERB) gumpalan gambar di `cmd-232' (NOUN) yang memiliki gumpalan yang terstruktur seperti satelit atau bintang (ADJECTIVE) dan tidak tumpang tindih dengan struktur gumpalan lain (ADJECTIVE)*

2.6 PROSES MENCOCOKKAN MASALAH DAN SOLUSI

SemanticServices. Net adalah layanan pencarian berbasis masalah yang mengambil masalah pengguna yang dijelaskan dalam SNL/SQDL dan mencoba mengidentifikasi satu atau lebih layanan yang kemampuannya dapat sesuai dengan deskripsi masalah. Penemuan layanan dapat dilakukan dalam dua fase: pendaftaran layanan diikuti oleh pencocokan layanan:

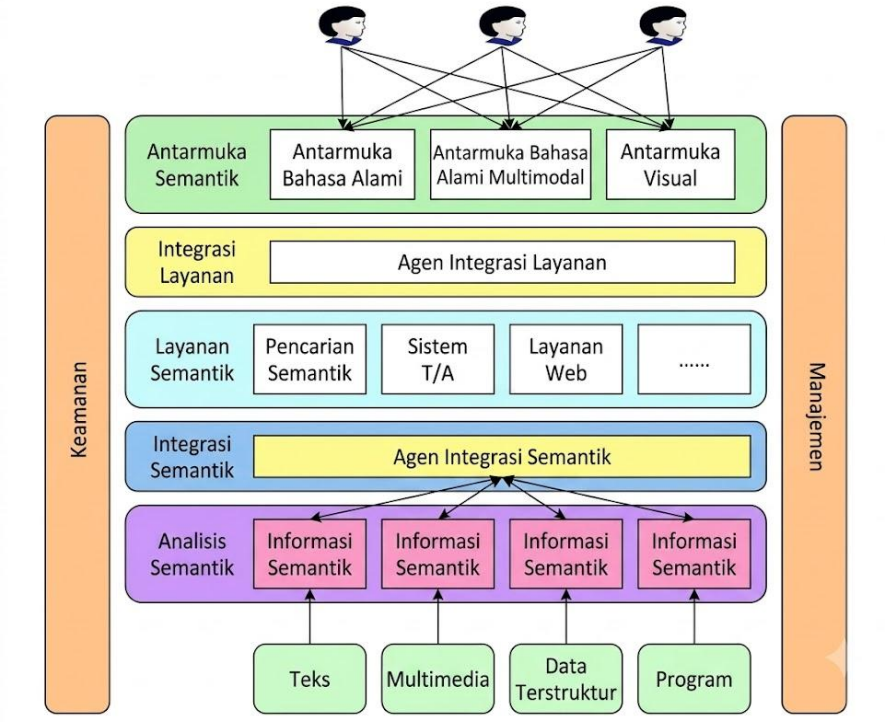
Pendaftaran Layanan: Agar dapat ditemukan, layanan harus mendaftar terlebih dahulu. Penyedia layanan harus menyediakan informasi layanan, termasuk URL layanan,

namespace, deskripsi SCDL, dll. Pencocokan Layanan: Ketika pengguna mengirimkan deskripsi SQDL (diterjemahkan dari kalimat SNL), Pencocok SQDL & SCDL menangani pencocokan antara deskripsi SQDL dan deskripsi SCDL yang tersedia.

Proses pencocokan terdiri dari dua bagian. Bagian pertama adalah pencocokan antarmuka — pencocok mencocokkan antarmuka deskripsi SQDL dengan antarmuka deskripsi SCDL dari setiap layanan yang terdaftar. Bagian kedua adalah pencocokan predikat — pencocok mencocokkan predikat deskripsi SQDL dengan predikat deskripsi SCDL dari setiap layanan yang terdaftar. Berdasarkan penyatuan yang tepat , pencocok menentukan apakah suatu layanan memiliki kemampuan untuk menyelesaikan masalah.

2.7 DEFINISI DAN RUANG LINGKUP KOMPUTASI SEMANTIK

Mesin pencari seperti yang dibahas di Bagian 3–6 merupakan contoh sistem Komputasi Semantik. Kami mendefinisikan "Komputasi Semantik" sebagai bidang yang membahas derivasi dan pencocokan semantik konten komputasional dan semantik niat pengguna yang diungkapkan secara alami untuk membantu mengambil, mengelola, memanipulasi, atau bahkan membuat konten, di mana "konten" dapat berupa apa saja termasuk video, audio, teks, proses, layanan, perangkat keras, jaringan, komunitas, dll. Ini menyatukan disiplin ilmu yang berkaitan dengan menghubungkan niat manusia (yang seringkali dirumuskan secara samar) dengan konten komputasional. Koneksi ini dapat berjalan dua arah: mengambil, menggunakan, dan memanipulasi konten yang ada sesuai dengan tujuan pengguna ("lakukan apa yang dimaksud pengguna"); dan membuat, mengatur ulang, dan mengelola konten yang sesuai dengan niat penulis ("lakukan apa yang dimaksud penulis").



Gambar 2.1 Arsitektur Komputasi Semantik.

Seperti yang ditunjukkan pada Gambar 2.1, sebagai revisi dari struktur yang dijelaskan, teknologi Komputasi Semantik dapat dikelompokkan menjadi lima lapisan:

(1) *Analisis Semantik*, yang menganalisis dan mengubah sinyal seperti piksel dan kata (konten) menjadi makna (semantik); ini menyediakan sumber daya informasi untuk Integrasi Semantik dan Aplikasi Semantik.

(2) *Integrasi Semantik*, yang mengintegrasikan konten dan semantik dari berbagai sumber dengan model terpadu; ini mempertimbangkan informasi yang diperoleh dari lapisan Analisis Semantik yang disajikan dalam berbagai format dan mengintegrasikan informasi tersebut sebelum dapat digunakan.

(3) *Layanan Semantik*, yang memanfaatkan konten dan semantik untuk menyelesaikan masalah tertentu. Layanan semantik telah dikembangkan dalam beberapa bentuk: Pencarian Web, termasuk Tanya Jawab (Q/A) otomatis dan pengambilan informasi; Basis Data Multimedia, dengan fokus utama pada pengambilan berbasis konten; dan sintesis semantik, yang merupakan kebalikan dari analisis semantik;

(4) *Integrasi Layanan*, yang mengintegrasikan berbagai layanan untuk menyediakan layanan yang lebih kuat. Integrasi layanan menyediakan metode interoperabilitas antara berbagai layanan.

(5) *Antarmuka Semantik*, yang memungkinkan pengguna untuk mengakses dan memanipulasi konten dan semantik dari berbagai sumber. Teknik Antarmuka Pengguna Grafis (GUI) standar seperti penjelajahan, pohon menu, dan bantuan daring jauh kurang menarik untuk aplikasi generasi berikutnya. Oleh karena itu, standar antarmuka baru, seperti antarmuka bahasa alami, antarmuka bahasa alami multimodal, serta antarmuka visual, menjadi semakin penting.

Selain lima lapisan yang dibahas di atas, arsitektur tersebut berisi mekanisme keamanan dan manajemen. Karena sebagian besar informasi dikirim dan diterima melalui jaringan, mekanisme keamanan di berbagai tingkatan diperlukan. Lapisan manajemen menyediakan manajemen keseluruhan arsitektur komputasi semantik. Lapisan ini mengontrol informasi apa yang diberikan untuk analisis semantik dan bagaimana informasi semantik dapat diintegrasikan untuk membangun gudang pengetahuan. Lapisan ini juga mengawasi resolusi konflik semantik antara berbagai sumber informasi. Pada tingkat teratas, ia mengelola interoperabilitas antara berbagai layanan dan antara berbagai antarmuka. Mesin pencari yang dibahas dalam artikel ini adalah contoh arsitektur Komputasi Semantik. Ia menggunakan SCDL sebagai bahasa terpadu untuk mendokumentasikan semantik (kemampuan) solusi dan mengintegrasikannya dengan model terpadu — Layanan. SNL/SQDL di sisi lain menyediakan antarmuka standar dan semantik untuk menafsirkan maksud pengguna.

2.8 EVALUASI KOMPOSISI LAYANAN DAN PROSPEKNYA

Untuk masalah kompleks yang tidak dapat dijawab oleh satu layanan tunggal, diharapkan *SemanticServices*.Net dapat mengidentifikasi serangkaian layanan yang secara kolektif dapat menyelesaikan masalah dan mensintesis alur kerja di antara layanan komponen.

Banyak pembelajaran terkait komposisi layanan web telah dilakukan untuk menyediakan platform dan bahasa untuk menyusun sistem heterogen, seperti *Universal Description, Discovery, and Integration* (UDDI), *Web Services Description Language* (WSDL), *Simple Object Access Protocol* (SOAP) dan bagian dari ontologi OWL-S (ServiceProfile dan ServiceGrounding). Platform dan bahasa tersebut mencoba mendefinisikan cara standar untuk penemuan, deskripsi, dan pemanggilan layanan (pengiriman pesan). Inisiatif lain seperti *Business Process Execution Language for Web Service* (BPEL4WS) dan OWL-S ServiceModel berfokus pada representasi alur kerja komposisi layanan. Dua teknik utama adalah komposisi berbasis aliran (seperti EFlow, bahasa definisi layanan komposit (CSDL), Model Proses Polimorfik (PPM) dan komposisi berbasis AI seperti kalkulus Situasi) dan perencanaan berbasis aturan.

Terlepas dari semua upaya ini, komposisi layanan otomatis masih memiliki jalan panjang yang harus ditempuh. Hambatan utama sintesis layanan otomatis adalah ketidaktersediaan detail layanan. Sementara WSDL dan OWL-S hanya mengekspos tanda tangan (yaitu, nama layanan, input dan output) dari suatu layanan dan tidak mengungkapkan sifat layanan tersebut, SCDL adalah bahasa orkestrasi yang cukup kaya untuk mengekspos detail layanan serta tanda tangannya. Pendekatan polinomial untuk sintesis otomatis layanan relasional dapat ditemukan.

Asumsi penting yang telah kami buat adalah bahwa konsumen dan penyedia berbagi terminologi yang sama untuk berkomunikasi melalui antarmuka SNL. Mempertimbangkan faktor 6 yang dibahas di Bagian. 2. Selain itu, organisasi yang netral dapat memainkan peran penting dalam memastikan platform komunikasi adil dan efisien. Kemajuan terbaru dalam Komputasi Utilitas dan Komputasi Awan dapat lebih memfasilitasi transisi Internet dari konten bersama ke pemecahan masalah, karena sejumlah besar lisensi perangkat lunak dapat dikonversi menjadi layanan perangkat lunak. *SemanticServices.Net,* sebagai infrastruktur dan implementasi Komputasi Semantik, dapat membantu komunitas Komputasi Awan untuk tumbuh lebih cepat. Terakhir, kami berharap layanan pemecahan masalah seperti yang dijelaskan dalam artikel ini dapat mencapai tujuan utama bahwa setiap layanan dapat diciptakan setara, dan kemampuannyalah yang akhirnya menentukan apakah layanan tersebut dapat diterima oleh pelanggan.

BAB 3
KLASIFIKASI TEKS BERBASIS PEMBOBOTAN RANDOM WALK

3.1 OPTIMASI PEMBOBOTAN ISTILAH MELALUI PENDEKATAN RANDOM WALK

Artikel ini menjelaskan pendekatan baru untuk memperkirakan bobot istilah dalam sebuah dokumen, dan menunjukkan bagaimana skema pembobotan baru dapat digunakan untuk meningkatkan akurasi pengklasifikasi teks. Metode ini menggunakan kemunculan bersama istilah sebagai ukuran ketergantungan antar fitur kata. Model random walk diterapkan pada grafik yang mengkodekan kata dan ketergantungan kemunculan bersama, menghasilkan skor yang mewakili kuantifikasi bagaimana fitur kata tertentu berkontribusi pada konteks tertentu. Eksperimen yang dilakukan pada tiga dataset klasifikasi standar menunjukkan bahwa pendekatan berbasis random walk yang baru mengungguli pendekatan frekuensi istilah tradisional dalam pembobotan fitur. Frekuensi istilah telah lama digunakan sebagai faktor utama untuk memperkirakan distribusi probabilitas fitur dalam sebuah dokumen, dan telah digunakan dalam berbagai tugas termasuk pemodelan bahasa, pemilihan fitur dan pembobotan istilah. Kelemahan utama yang terkait dengan metode frekuensi istilah adalah kenyataan bahwa metode ini bergantung pada pendekatan bag-of-words. Hal ini menyiratkan independensi fitur, dan mengabaikan ketergantungan apa pun yang mungkin ada antara kata-kata dalam teks. Dengan kata lain, ini mendefinisikan "pilihan acak," di mana bobot istilah sebanding dengan probabilitas memilih istilah secara acak dari himpunan istilah yang membentuk teks. Pendekatan seperti itu mungkin efektif untuk menangkap relevansi suatu istilah dalam konteks lokal, tetapi gagal memperhitungkan efek global yang ditimbulkan oleh keberadaan istilah tersebut pada seluruh segmen teks.

Kami berpendapat bahwa model *bag-of-words* mungkin bukan teknik terbaik untuk menangkap pentingnya istilah. Sebaliknya, mengingat bahwa relasi dalam teks dapat dipertahankan dengan menjaga representasi struktural teks, metode yang memperhitungkan sifat struktural konteks dapat menghasilkan skema pembobotan istilah yang lebih baik. Pembelajaran sebelumnya telah menunjukkan bahwa kinerja yang lebih tinggi tetapi mahal dapat dicapai dengan menggabungkan ketergantungan tersebut. Dalam artikel ini, kami memperkenalkan sistem yang memodelkan masalah pembobotan sebagai "jalan acak" daripada "pilihan acak." Kami mengasumsikan pembaca imajiner (atau "pejalan") yang melangkah melalui teks berdasarkan istilah demi istilah. Dalam pengaturan ini, pentingnya istilah ditentukan oleh probabilitas penjelajah acak untuk menemukan istilah target dalam teks selama penjelajahan.

Ukuran baru pembobotan istilah mengintegrasikan baik lokalitas suatu istilah maupun hubungannya dengan konteks sekitarnya. Kami memodelkan kontribusi lokal ini menggunakan relasi ko-okurensi di mana istilah-istilah yang muncul bersamaan dalam konteks tertentu cenderung berbagi beberapa kepentingan (atau signifikansi) di antara mereka. Perhatikan bahwa dalam model ini, hubungan antara istilah tertentu dan konteksnya tidak linier. Suatu istilah tertentu berhubungan dengan suatu konteks, dan konteks tersebut, pada gilirannya,

berhubungan dengan kumpulan istilah. Untuk memodelkan relasi rekursif ini, kami menggunakan algoritma pemeringkatan berbasis grafik, yaitu algoritma penjelajahan acak PageRank, dan adaptasi TextRank-nya untuk pemrosesan teks. Dalam artikel ini, kami menunjukkan bagaimana TextRank dapat digunakan untuk memodelkan distribusi probabilitas fitur kata dalam sebuah dokumen. Melalui eksperimen yang dilakukan pada tugas klasifikasi teks, kami menunjukkan bahwa skor random walk mengungguli frekuensi istilah tradisional, yang biasanya digunakan untuk memodelkan bobot fitur untuk tugas ini.

Berikut ini, pertama-tama kami mengulas prinsip-prinsip dasar di balik algoritma random walk, dan secara singkat menjelaskan aplikasi TextRank untuk pemrosesan teks. Kemudian kami menunjukkan bagaimana model random walk ini dapat diadaptasi untuk pembobotan istilah, dan menunjukkan bahwa skema pembobotan baru dapat digunakan untuk secara signifikan meningkatkan akurasi sistem klasifikasi teks, dibandingkan dengan skema pembobotan frekuensi istilah tradisional. Terakhir, kami menyimpulkan dengan diskusi dan arahan untuk pekerjaan di masa mendatang.

3.2 ALGORITMA RANDOM WALK

Ide dasar yang diimplementasikan oleh algoritma random walk adalah "pemungutan suara" atau "rekomendasi." Ketika satu simpul terhubung ke simpul lain, pada dasarnya ia memberikan suara untuk simpul tersebut. Semakin tinggi jumlah suara yang diberikan untuk suatu simpul, semakin tinggi pentingnya simpul tersebut. Selain itu, pentingnya simpul yang memberikan suara menentukan seberapa penting suara itu sendiri, dan informasi ini juga diperhitungkan oleh algoritma pemeringkatan. Meskipun ada beberapa algoritma random walk yang telah diusulkan di masa lalu, kami hanya fokus pada satu algoritma tersebut, yaitu PageRank, karena sebelumnya telah terbukti berhasil dalam sejumlah aplikasi, termasuk analisis tautan web, jaringan sosial, analisis kutipan, dan baru-baru ini dalam beberapa aplikasi pemrosesan teks. Diberikan sebuah graf $G = (V, E)$, misalkan $In(V_a)$ adalah himpunan simpul yang menunjuk ke simpul V_a (pendahulu), dan $Out(V_a)$ adalah himpunan simpul yang ditunjuk oleh simpul V_a (penerus). Skor PageRank yang terkait dengan simpul Va didefinisikan menggunakan fungsi rekursif yang mengintegrasikan skor pendahulunya:

$$S(V_a) = (1 - d) + d * \sum_{V_b \in In(V_a)} \frac{S(V\_b).}{Out(V_b)} \quad (3.1)$$

di mana d adalah parameter yang nilainya ditetapkan antara 0 dan 1.[a] Skor setiap simpul dihitung ulang pada setiap iterasi berdasarkan bobot baru yang telah dikumpulkan oleh simpul-simpul tetangga. Algoritma berakhir ketika titik konvergensi tercapai untuk semua simpul, yang berarti bahwa tingkat kesalahan untuk setiap simpul berada di bawah ambang batas yang telah ditentukan.

Skema penilaian simpul ini didasarkan pada model random walk, di mana seorang pejalan mengambil langkah acak pada grafik, dengan pergerakan tersebut dimodelkan sebagai proses Markov. Dalam kondisi tertentu (grafik bersifat aperiodik dan tidak dapat direduksi),

model tersebut dijamin akan konvergen ke distribusi stasioner probabilitas yang terkait dengan simpul-simpul dalam grafik. Secara intuitif, probabilitas stasioner yang terkait dengan suatu simpul mewakili probabilitas menemukan pejalan di simpul tersebut selama random walk, dan dengan demikian mewakili pentingnya simpul tersebut dalam grafik.

Yang sangat relevan untuk pekerjaan kami adalah penerapan random walk pada pemrosesan teks, seperti yang dilakukan dalam sistem TextRank. TextRank telah berhasil diterapkan pada tiga tugas pemrosesan bahasa alami: peringkasan dokumen, disambiguasi makna kata, dan ekstraksi kata kunci, dengan hasil yang kompetitif dengan sistem canggih terkini. Kekuatan model ini terletak pada representasi global konteks dan kemampuannya untuk memodelkan bagaimana kemunculan bersama antar fitur dapat menyebar di seluruh konteks dan memengaruhi fitur-fitur lain yang jauh. Pendekatan kami mengikuti langkah-langkah serupa seperti yang digunakan dalam aplikasi ekstraksi kata kunci TextRank, yang menurunkan bobot istilah menggunakan representasi grafik yang memperhitungkan ketergantungan kemunculan bersama antar kata dalam teks. Namun, kami menggabungkan sejumlah unit leksikal yang lebih besar, dan menggunakan ukuran jendela yang berbeda, seperti yang akan kami tunjukkan di bagian selanjutnya.

3.3 RANDOM WALK UNTUK PEMBOBOTAN ISTILAH

Dimulai dengan dokumen tertentu, kami menentukan peringkat atas kata-kata dalam dokumen dengan menggunakan model-model berikut.

Model random walk

Dalam pekerjaan kami, kami bereksperimen dengan beberapa variasi PageRank yang menggabungkan informasi dan variabel tambahan ke dalam versi tradisional yang ditunjukkan pada Persamaan (3.1). Kami merangkum model pemeringkatan istilah berbasis PageRank terbaik sebagai berikut:

$rwo \overset{\leftrightarrow}{o}$: Ini mewakili model asli, seperti yang dijelaskan dalam Persamaan (3.1) di mana kami menggunakan grafik tak berarah dengan faktor redaman konstan yang secara ketat mengikuti rumus tradisional PageRank.

$rw^{\leftrightarrow}_{e.idf}$: Model ini mewakili pendekatan grafik tak berarah yang menggunakan versi tepi berbobot PageRank dengan faktor redaman variabel. Bobot tepi dihitung dengan rumus berikut:

$$E_{V_1}, V_2 = tf.idf_{V_1} * tf.idf_{V_2}. \quad (3.2)$$

di mana E_{V_1}, V_2 adalah sisi yang menghubungkan V_1 ke V_2, dan $tf.idf$ mewakili frekuensi istilah dikalikan dengan frekuensi dokumen invers. Faktor redaman dinyatakan sebagai fungsi dari bobot sisi yang masuk, dihitung sebagai berikut:

$$d_{E_{V_1V_2}} = E_{V_1,V_2}/E_{max.} \quad (3.3)$$

di mana $d_{E_{V_1}}$,$_{V_2}$ adalah fungsi redaman dan $E_{\ max}$ mewakili bobot tertinggi untuk sebuah sisi dalam grafik. Rumus pemeringkatan simpul yang dihasilkan adalah:

$$S'(V_a) = \frac{(1\ -\ d)}{|N|} + \sum_{V_b \in In(V_a)} C\ *\ \frac{d_{E_{V_b}}, V_a * S(V_b)}{|Out(V_b)|}. \tag{3.4}$$

di mana N mewakili jumlah total node dalam grafik, d adalah konstanta redaman, dan C adalah konstanta penskalaan.^{b} Model ini membiaskan random walker ke arah node dengan edge yang lebih kuat (bobot lebih tinggi), dibandingkan dengan node dengan edge yang lebih lemah (bobot lebih rendah).

$rw_{e.o\overleftrightarrow{c}}$: Model ini mirip dengan pendekatan di atas, namun faktor redaman untuk sebuah edge diperkirakan berdasarkan frekuensi kemunculan bersama bigram dari dua node yang dihubungkan oleh edge tersebut, Persamaan (3.5). Misalnya, jika bigram "free software" muncul empat kali dalam sebuah dokumen, maka bobot edge yang menghubungkan "free" dan "software" adalah empat.

$$E_{V_1}, V_2 =\ tf(v_1 v_2). \tag{3.5}$$

Pembobotan Istilah

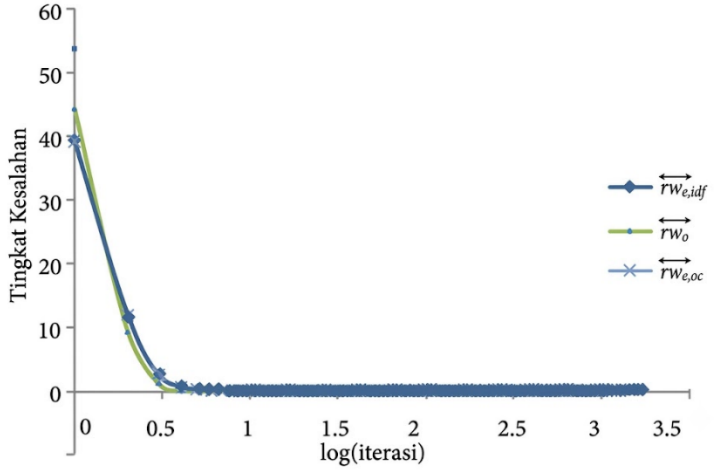
Dengan diberikan sebuah dokumen, langkah-langkah berikut diterapkan untuk mendapatkan bobot yang terkait dengan kata-kata dalam teks: Pertama, dokumen dipecah menjadi token untuk tanda baca, simbol khusus, dan singkatan kata. Kata-kata umum juga dihilangkan, menggunakan daftar sekitar 500 kata yang sering digunakan.

Selanjutnya, teks yang dihasilkan diproses untuk mengekstrak bobot frekuensi istilah (tf) dan random walk (rw) untuk setiap istilah dalam dokumen. Untuk menentukan tf, kita cukup menghitung frekuensi setiap kata dalam dokumen. Untuk menentukan rw, semua istilah ditambahkan sebagai simpul dalam grafik yang mewakili dokumen. Pemindai ko-okurensi kemudian diterapkan pada teks untuk menghubungkan istilah-istilah yang muncul bersamaan dalam ukuran jendela tertentu. Untuk suatu istilah tertentu, semua istilah yang berada di sekitar istilah ini dianggap sebagai istilah dependen. Ini diwakili oleh serangkaian tepi yang menghubungkan istilah tersebut ke semua istilah lain dalam jendela. Eksperimen dilakukan untuk ukuran jendela 2, 4, 6, dan 8. Setelah grafik dibangun dan sisi-sisinya ditempatkan, algoritma random walk diterapkan. Hasil pemeringkatan adalah daftar semua istilah masukan dan skor rw yang sesuai.

Sebuah contoh

Untuk memahami mengapa bobot rw mungkin merupakan pengganti yang baik untuk bobot tf tradisional, pertimbangkan contoh pada Gambar 3.1 , yang memodelkan dokumen sampel. Dimulai dengan teks ini, grafik dibangun sebagai berikut. Jika suatu istilah belum

pernah dilihat sebelumnya, maka sebuah node ditambahkan ke grafik untuk mewakili istilah ini.

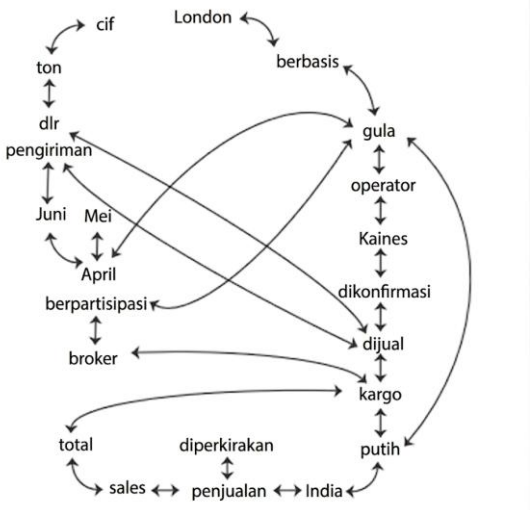


Gambar 3.1 Grafik konvergensi untuk model random walk.

Operator gula yang berbasis di London, Kaines Ltd, mengkonfirmasi telah menjual dua kargo gula putih ke India dari total penjualan sekitar empat atau lima kargo yang melibatkan broker lain. Gula tersebut, untuk pengiriman April/Mei dan April/Juni, dijual dengan harga antara 214 dan 218 dolar per ton CIF, katanya.

Gambar 3.2 Contoh dokumen.

Suatu istilah hanya dapat diwakili oleh satu node dalam grafik. Sebuah sisi tak berarah digambar antara dua node jika keduanya muncul bersamaan dalam ukuran jendela tertentu. Gambar 3.3 menunjukkan grafik yang dibangun untuk teks ini, dengan asumsi ukuran jendela 2, yang sesuai dengan dua istilah berurutan dalam teks (misalnya, London dihubungkan dengan based).



Gambar 3.3 Contoh grafik.

Tabel 3.1 Skor tf & rw untuk contoh teks.

Istilah	rw	tf	Istilah	rw	tf
gula	16.88	3	berpartisipasi	3.87	1
dijual	14.15	2	april	3.87	2
berdasarkan	7.39	1	india	1.00	1
dikonfirmasi	6.90	1	diperkirakan	1.00	1
Kaines	6.81	1	penjualan	1.00	1
operator	6.76	1	total	1.00	1
London	4.14	1	pialang	1.00	1
kargo	4.01	2	mei	1.00	1
pengiriman	4.01	1	juni	1.00	1
dolar	4.01	1	ton	1.00	1
putih	3.87	1	cif	1.00	1

Setelah grafik dibangun, model random walk diterapkan pada grafik, menghasilkan serangkaian skor yang terkait dengan simpul (kata). Tabel 3.1 menunjukkan bobot tf dan rw. Dengan menganalisis bobot rw, kita dapat mengamati korelasi non-linear dengan bobot tf, dengan penekanan pada istilah-istilah di sekitar istilah kunci penting seperti gula atau kargo. Lokalisasi spasial ini menghasilkan peringkat yang lebih tinggi untuk istilah seperti operator dibandingkan dengan istilah lain seperti London.

3.4 PENGATURAN EKSPERIMENTAL

Untuk mengevaluasi pendekatan berbasis random walk untuk pembobotan fitur, kami mengintegrasikannya ke dalam algoritma klasifikasi teks, dan mengevaluasi kinerjanya pada beberapa dataset klasifikasi teks standar. Klasifikasi teks adalah masalah yang biasanya dirumuskan sebagai tugas pembelajaran mesin, di mana pengklasifikasi mempelajari cara membedakan antara kategori dalam himpunan tertentu dengan menggunakan fitur yang diekstrak secara otomatis dari kumpulan dokumen pelatihan.

Pengklasifikasi Teks

Kami membandingkan hasil yang diperoleh dengan tiga pengklasifikasi teks yang sering digunakan — Rocchio, Naïve Bayes, dan SVM, yang dipilih berdasarkan kinerja dan keragaman metodologi pembelajarannya.

Naïve Bayes. Ide dasar dalam pengklasifikasi teks Naïve Bayes adalah untuk memperkirakan probabilitas suatu kategori berdasarkan dokumen menggunakan probabilitas gabungan kata dan dokumen. Pengklasifikasi teks Naïve Bayes mengasumsikan independensi kata, tetapi terlepas dari penyederhanaan ini, mereka terbukti berkinerja sangat baik. Meskipun terdapat beberapa versi pengklasifikasi Naïve Bayes (variasi multinomial dan Bernoulli multivariat), kami menggunakan model multinomial, yang terbukti lebih efektif.

Rocchio. Metode klasifikasi teks Rocchio menggunakan vektor berbobot istilah standar untuk merepresentasikan dokumen, dan membangun vektor prototipe untuk setiap kategori dengan menjumlahkan vektor dokumen pelatihan di setiap kategori. Dokumen uji kemudian ditugaskan ke kategori dengan vektor prototipe terdekat, berdasarkan kesamaan kosinus. Eksperimen klasifikasi dengan berbagai versi algoritma menunjukkan hasil yang kompetitif pada benchmark standar.

SVM. Support Vector Machines (SVM) adalah pendekatan pembelajaran mesin canggih berdasarkan rencana keputusan. Algoritma ini mendefinisikan hyper-plan terbaik yang memisahkan himpunan titik yang terkait dengan label kelas yang berbeda dengan margin maksimum. Contoh yang tidak berlabel kemudian diklasifikasikan dengan memutuskan di sisi mana dari hyper-surface mereka berada. Dalam evaluasi kami, kami menggunakan $SVMTorch$ dengan kernel linier, karena terbukti sama kuatnya dengan kernel lain dalam eksperimen klasifikasi teks. Implementasi SVM ini juga diamati sebagai yang tercepat jika dibandingkan dengan $SVMlib$ dan $SMO\ Weka$.

Kami menggunakan bobot fitur tf dan rw (dan alternatifnya, seperti yang dijelaskan di bawah) untuk membuat vektor fitur untuk pengklasifikasi SVM dan Naïve Bayes. Mengikuti praktik standar dalam pembobotan istilah untuk pengklasifikasi Rocchio, kami menggunakan bobot fitur $tf.idf$ dan $rw.idf$ dalam evaluasi awal model kami (Tabel 2.2, 3.3, 3.5 dan 3.6). Dalam eksperimen akhir (Tabel 3.8 dan 3.9), kami melaporkan hasil tf dan $tf.idf$ secara terpisah. Hasil yang diperoleh menggunakan tf akan bertindak sebagai dasar untuk semua evaluasi.

Kumpulan Data

Kami menggunakan tiga kumpulan data standar: *WebKB, LingSpam,* dan 20*Newsgroups* — yang umum digunakan dalam evaluasi klasifikasi teks.

WebKB [g] adalah kumpulan data yang dikumpulkan dari departemen ilmu komputer di berbagai universitas. Kumpulan data ini berisi tujuh label kelas: Proyek, Mahasiswa, Departemen, Fakultas, Staf, Mata Kuliah, dan Lainnya. Label Lainnya dihapus dari kumpulan data untuk tujuan evaluasi. Sebagian besar evaluasi dalam riset hanya dilakukan pada empat kategori (Proyek, Mahasiswa, Fakultas, dan Mata Kuliah) karena kategori tersebut merupakan kategori terbesar. Namun, karena kami ingin melihat bagaimana sistem kami berperilaku ketika hanya sedikit contoh pelatihan yang tersedia, kami juga mempertimbangkan kelas Staf dan Departemen yang hanya memiliki sedikit dokumen pelatihan yang tersedia. Kami melakukan evaluasi pada dua versi $WebKB$: satu dengan versi empat kategori ($WebKB_4$) dan satu dengan enam kategori ($WebKB_6$).

20Newsgroups [h] adalah kumpulan 20.000 pesan dari 20 newsgroup, yang sesuai dengan berbagai topik atau subjek. Setiap newsgroup memiliki sekitar 1000 pesan yang dibagi menjadi 400 dokumen uji dan 600 dokumen pelatihan.

LingSpam [i] adalah korpus spam, yang terdiri dari pesan email yang diorganisir dalam 10 set untuk memungkinkan validasi silang 10-fold**.** Setiap koleksi memiliki sekitar 300 pesan spam dan pesan yang sah. Terdapat empat versi korpus yang mewakili versi polos, versi yang disaring stop-word, versi yang dilematisasi, dan versi stop-word dan

yang dilematisasi. Kami menggunakan koleksi polos dengan validasi silang 10-fold standar.

3.5 EVALUASI MODEL RANDOM WALK UNTUK KLASIFIKASI TEKS

Sebagai langkah pertama, kami mengevaluasi setiap model random walk yang disajikan di Bagian 3 ($rw_o^{\leftrightarrow}, rw_{e.idf}^{\leftrightarrow}, dan\ rw_{e.oc}^{\leftrightarrow}$). Tabel 3.2 dan 3.3 menunjukkan angka akurasi rata-rata mikro dan rata-rata makro untuk setiap model, pengklasifikasi, dan dataset untuk ukuran jendela 2. Hasil ini juga ditampilkan secara grafis pada Gambar 3.4 dan 3.5. Kolom tf menunjukkan hasil yang diperoleh menggunakan skema pembobotan frekuensi istilah, yang, seperti yang dinyatakan sebelumnya, bertindak sebagai dasar di seluruh eksperimen kami. Seperti yang terlihat pada tabel, sebagian besar model yang disajikan berkinerja lebih baik daripada baseline tf. Baik model $rw_{e.oc}^{\leftrightarrow}$ dan $rw_{e.idf}^{\leftrightarrow}$ menonjol sebagai model berkinerja terbaik dengan peningkatan yang nyata, terutama untuk pengklasifikasi SVM dan Rocchio.

Tabel 3.2 Hasil rata-rata mikro untuk berbagai model random walk.

Kumpulan Data	tf	$rw_{e.oc}^{\leftrightarrow}$	$rw_{e.idf}^{\leftrightarrow}$	$rw_o^{\leftrightarrow}$
Naïve Bayes				
$WebKB_4$	84.2	**86.1***	85.8*	86.1**
$WebKB6$	81.3	82.4	81.9	**83.3***
$LSpam$	99.2	**99.3**	99.2	99.3
$20NG$	89.3	91.5**	**91.7****	90.6**
$Rocchio^{idf}$				
$WebKB_4$	84.3	87.5**	**87.7****	86.9**
$WebKB_6$	80.1	**84.3****	84.2**	83.4**
$LSpam$	98.1	**98.5**	98.3	98.2
$20NG$	91.5	**94.3****	93.8**	93.0**
SVM				
$WebKB_4$	81.3	**94.1****	93.2**	80.3
$WebKB_6$	79.3	**90.7****	90.4**	78.9
$LSpam$	93.6	98.9**	**99****	92.1
$20NG$	90.1	**94.4****	94.4**	86.9

Tabel 3.3 Hasil rata-rata makro untuk berbagai model random walk.

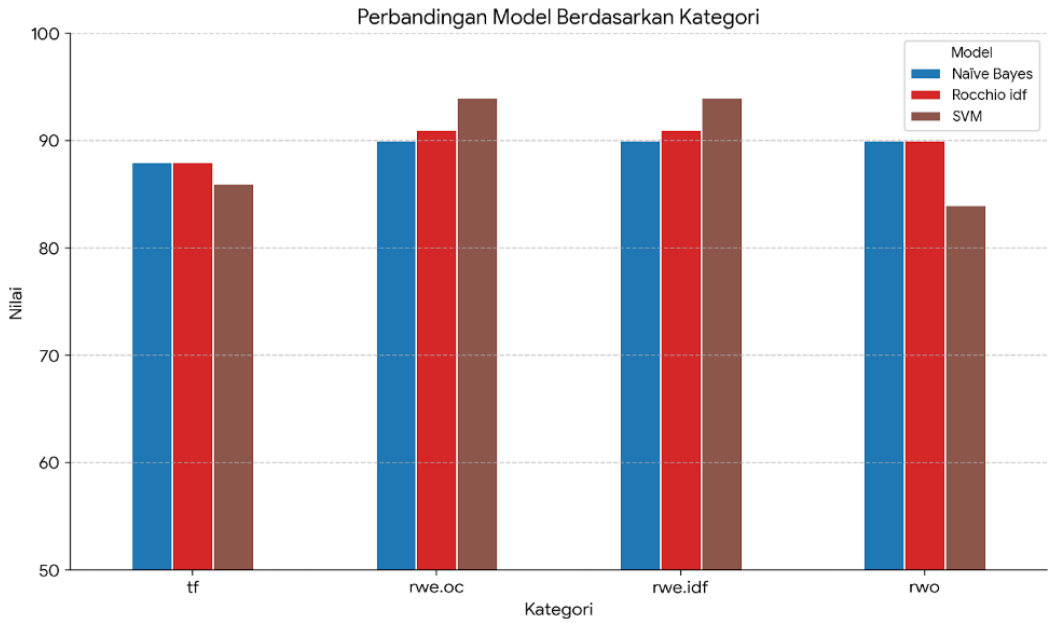
Kumpulan Data	tf	$rw_{e.oc}^{\leftrightarrow}$	$rw_{e.idf}^{\leftrightarrow}$	$rw_o^{\leftrightarrow}$
Naïve Bayes				
$WebKB_4$	82.5	83.5*	82.9*	**84.2****
$WebKB_6$	68.1	66.5	65.5	**70.0***
$LSpam$	98.6	**98.8**	98.6	98.8
$20NG$	89.3	**91.4****	90.6**	90.5**
$Rocchio^{idf}$				
$WebKB_4$	84.3	**86.7****	86.7**	86.1**
$WebKB_6$	71.6	**75.8****	75.3**	75.1**
$LSpam$	96.7	**97.3**	97.1	96.9

$20NG$	91.5	**94.3****	93.7**	93.0**
SVM				
$WebKB_4$	77.4	**93.2****	93.2**	80.3
$WebKB_6$	68.2	81.1**	**81.5****	65.2
$LSpam$	86.3	97.9**	**98.1****	81.2
$20NG$	90.6	**94.4****	94.4**	88.1

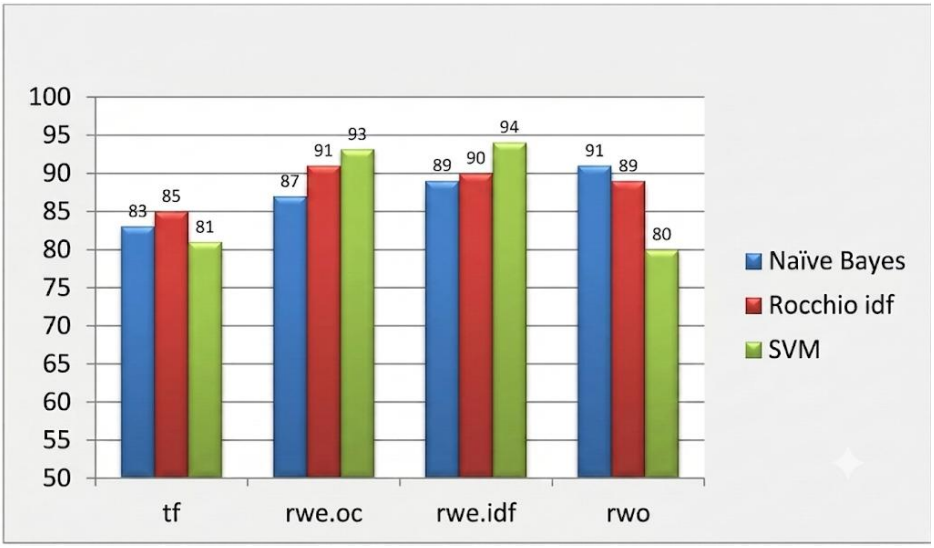
Tabel 3.4 Waktu eksekusi dalam detik untuk pemrosesan 1000 dokumen dari WebKB4, pada mesin Pentium-IV dengan RAM 2Gb.

tf	$rw^{\leftrightarrow}_{e.oc}$	$rw^{\leftrightarrow}_{e.idf}$	$rw^{\leftrightarrow}_{o}$
112	138	138	165

Model $rw^{\leftrightarrow}_{e.idf}$ dan $rw^{\leftrightarrow}_{e.oc}$ mendefinisikan ulang komponen lompatan acak Page-Rank, dengan mempertimbangkan faktor peredaman sebagai fungsi yang dapat diperkirakan per tepi. Sebuah node yang sangat terhubung dengan tepi yang relatif kuat cenderung mendorong penjelajah acak untuk mengikuti tautan keluarnya daripada melompat secara acak. Pertimbangan yang diberikan pada bobot relatif tepi menandakan enkapsulasi informasi global dalam faktor bias. Ini memungkinkan kita, dalam arti tertentu, untuk mengarahkan penjelajah acak ke node yang berguna secara lebih efektif, yang berharga dalam menekankan kekuatan diskriminatif fitur sentral.



Gambar 3.4 Hasil rata-rata mikro keseluruhan untuk berbagai model random walk.



Gambar 3.5 Hasil rata-rata makro keseluruhan untuk berbagai model random walk.

Selain akurasi, kami juga mengevaluasi efisiensi model baru tersebut. Dengan membandingkan waktu pemrosesan untuk 1000 dokumen *WebKB4* menggunakan model yang diusulkan, kami melihat sedikit peningkatan waktu pemrosesan sebesar 53 detik untuk model $rw_o^{\leftrightarrow}$ dan 26 detik untuk $rw_{e.oc}^{\leftrightarrow}$ dan $rw_{e.idf}^{\leftrightarrow}$, dibandingkan dengan baseline tf (yang juga mencakup waktu yang dibutuhkan untuk tokenisasi dan penghapusan stopword). Hal ini disebabkan oleh konvergensi cepat model-model ini dalam sekitar 15 iterasi (Gambar 3.1). Kami percaya bahwa peningkatan kecil dalam waktu pemrosesan ini merupakan biaya yang wajar untuk mencapai akurasi yang jauh lebih tinggi.

Ukuran Jendela yang Berbeda

Di antara berbagai model, model $rw_{e.oc}^{\leftrightarrow}$ tampaknya secara konsisten mengungguli model lainnya. Untuk melihat lebih dekat model ini, kami menganalisisnya lebih lanjut di bawah ukuran jendela yang berbeda. Tabel 3.5 dan 3.6 menunjukkan hasil klasifikasi $rw_{e.oc}^{\leftrightarrow}$ untuk $WebKB_4$, $WebKB_6$, *LingSpam*, dan *20Newsgroups* masing-masing. rw_2, rw_4, rw_6, dan rw_8 mewakili akurasi yang dicapai menggunakan skema pembobotan $rw_{e.oc}^{\leftrightarrow}$ di bawah ukuran jendela 2, 4, 6, dan 8 masing-masing.

Tabel 3.5 Hasil rata-rata mikro untuk model random walk $rw_{e.oc}^{\leftrightarrow}$ untuk ukuran jendela yang berbeda.

Dataset	tf	rw_2	rw_4	rw_6	rw_8
$Naïve\ Bayes$					
$WebKB_4$	84.2	**86.1***	85.8*	85.8*	85.7*
$WebKB_6$	81.3	**82.4***	81.9*	81.8	76.6
$LSpam$	99.2	99.3	**99.3**	99.3	99.3
$20NG$	89.3	**91.5**	91.2	91.2	91.2
$Rocchio^{idf}$					
$WebKB_4$	84.3	87.5**	87.5**	87.4**	**87.6****
$WebKB_6$	80.1	84.3**	84.0**	84.3**	**84.4****
$LSpam$	98.1	**98.5**	98.3	98.4	98.3

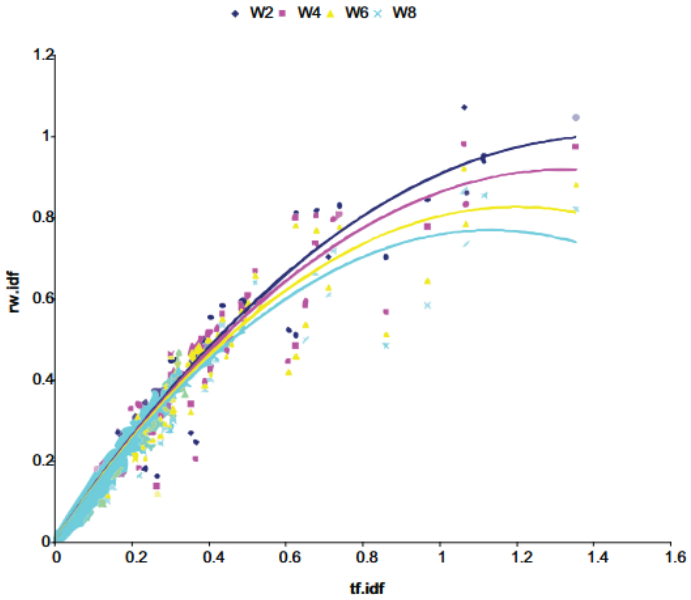
$20NG$	91.5	94.3**	**94.3****	94.2**	94.2**
SVM					
$WebKB_4$	81.3	**94.1****	93.6**	93.5**	93.7**
$WebKB_6$	79.3	**90.7****	90.6**	90.7**	90.6**
$LSpam$	93.6	98.9**	**99.1****	99.1**	99.0**
$20NG$	90.1	94.4**	94.5**	94.5**	**94.6****

Tabel 3.6 Hasil rata-rata makro untuk model random walk $rw^{\leftrightarrow}_{e.oc}$ untuk berbagai ukuran jendela.

Dataset	$\boldsymbol{tf}$	$\boldsymbol{rw}_2$	$\boldsymbol{rw}_4$	$\boldsymbol{rw}_6$	$\boldsymbol{rw}_8$
$Naïve\ Bayes$					
$WebKB_4$	82.5	**83.5***	83.0*	82.9*	82.6*
$WebKB_6$	68.1	**66.5***	66.0*	65.6	62.1
$LSpam$	98.6	98.8	**98.8**	98.8	98.8
$20NG$	89.3	**91.4**	91.2	91.1	91.1
$Rocchio^{idf}$					
$WebKB_4$	83.4	**86.7****	86.6**	86.5**	86.7**
$WebKB_6$	71.6	75.8**	75.5**	75.7**	**76.2****
$LSpam$	96.7	**97.3**	97.1	97.3	97.1
$20NG$	91.5	**94.3****	94.2**	94.2**	94.2**
SVM					
$WebKB_4$	77.4	**93.2****	92.9**	92.7**	92.9**
$WebKB_6$	68.2	81.1**	**81.5****	81.3**	81.3**
$LSpam$	86.3	97.9**	98.3**	**98.4****	98.2**
$20NG$	90.6	94.4**	94.5**	94.5**	**94.6****

Sistem ini menunjukkan kinerja yang konsisten di berbagai ukuran jendela. Dengan menganalisis lebih lanjut hasilnya menggunakan uji-t statistik, kami melihat bahwa jendela berukuran 2 dan 4 memberikan hasil yang paling signifikan di semua pengklasifikasi dan dataset. Membandingkan skema pembobotan tf dan rw untuk dataset $WebKB_6$, kami menemukan bahwa kedua skema tersebut gagal memprediksi kelas Staf. Namun, peningkatan yang signifikan diperoleh pada kelas Departemen, di mana model rw kami mencetak akurasi 47% dibandingkan dengan 4% saat menggunakan tf. Hal ini mungkin disebabkan oleh kemampuan model untuk mengekstrak distribusi istilah yang lebih realistis dan lebih halus, sehingga mengurangi bias fitur yang disebabkan oleh jumlah contoh pelatihan yang terbatas.

Kami juga melihat dampak positif dari penggunaan model kami dengan pengklasifikasi Rocchio. Untuk mencapai pemahaman yang lebih dalam tentang korelasi antara kedua model dalam konteks pengklasifikasi Rocchio, kami melakukan averaging makro terhadap skor istilah $tf.idf$ dan $rw.idf$ di seluruh dokumen *WebKB4*, dengan cara yang mirip dengan pembuatan vektor klasifikasi prototipe, tetapi dengan mempertimbangkan seluruh korpus daripada kelas individual. Skor istilah yang dihasilkan diplot pada Gambar 3.6 Aproksimasi polinomial digunakan untuk memvisualisasikan tren titik-titik yang diplot, di mana setiap garis tren mewakili ukuran jendela yang berbeda.



Gambar 3.6 Korelasi model tf dan rw untuk koleksi WebKB4.

Dengan menganalisis grafik, kita dapat membedakan tiga sifat menarik:

(1) Korelasi non-linier yang jelas antara model tf dan rw,
(2) Peningkatan penyimpangan dari model tf seiring dengan peningkatan ukuran jendela,
(3) Efek penghalusan yang terkait dengan model rw, dengan tingkat pertumbuhan nilai tf yang jelas lebih cepat daripada rw.

Tabel 3.7 Korelasi antara bobot fitur tf dan rw, untuk berbagai ukuran jendela.

Dataset	rw_2	rw_4	rw_6	rw_8
Naïve Bayes				
$WebKB_4$	0.80	**0.81**	0.80	0.79
$WebKB_6$	**0.83**	0.80	0.81	0.80
$LSpam$	**0.84**	0.84	0.84	0.84
$20NG$	**0.85**	0.84	0.83	0.83
$Rocchio^{idf}$				
$WebKB_4$	**0.73**	0.71	0.69	0.69
$WebKB_6$	**0.74**	0.71	0.70	0.68
$LSpam$	**0.82**	0.82	0.82	0.82
$20NG$	**0.62**	0.54	0.56	0.59
SVM				
$WebKB_4$	**0.48**	0.43	0.41	0.42
$WebKB_6$	**0.53**	0.52	0.51	0.52
$LSpam$	0.71	**0.73**	0.72	0.71
$20NG$	0.62	0.66	**0.67**	0.66

Kami juga memeriksa korelasi prediksi antara model *tf* dan *rw* sehubungan dengan pengklasifikasi dan ukuran jendela yang berbeda. Korelasi, yang ditunjukkan pada Tabel 3.7,

dihitung menggunakan metrik korelasi Pearson, dan sekali lagi menunjukkan korelasi yang lemah antara kedua model tersebut. Kedua model tersebut umumnya lebih beragam dan kurang berkorelasi ketika menggunakan jendela berukuran 6 dan 8, dibandingkan dengan jendela berukuran 2 dan 4. Hal ini disebabkan oleh peningkatan penyimpangan dari asumsi independensi fitur yang tersirat oleh *tf*. Namun, peningkatan ketergantungan tidak selalu diinginkan, seperti yang terlihat pada hasil yang dilaporkan. Kami memperkirakan bahwa pada ukuran jendela tertentu, kinerja sistem akan menurun menjadi skema pembobotan biner, untuk ukuran jendela yang sama dengan ukuran dokumen. Dalam kasus seperti itu, setiap istilah akan bergantung pada semua istilah yang tersisa sehingga menghasilkan grafik yang hampir sepenuhnya terhubung. Akibatnya, setiap kontribusi fitur terhadap konteks sekitarnya akan identik, sehingga menghasilkan skor *rw* yang serupa untuk semua fitur.

Kebijakan skema pembobotan lainnya

Kami juga membandingkan model kami dengan skema pembobotan canggih lainnya yang dilaporkan:

$tf.idf$: sejak diperkenalkan, $tf.idf$ telah menjadi salah satu skema pembobotan yang paling banyak dipelajari. Skema ini berfungsi sebagai standar dasar dalam studi pembobotan istilah dan terbukti sulit dikalahkan.

itf: didefinisikan sebagai satu minus frekuensi istilah invers, skema pembobotan ini ditemukan memiliki kinerja yang sangat baik ketika digunakan dengan kernel linier SVM.

$log(tf)$: pertama kali diperkenalkan, skema ini baru-baru ini digunakan dengan tujuan menghaluskan frekuensi istilah dan karenanya meminimalkan bias fitur.

$log(tf).idf$: skema ini disarankan, dan menunjukkan kinerja yang unggul. Dalam skema ini, frekuensi istilah yang dihaluskan diskalakan oleh idf-nya untuk memberikan bobot yang lebih tinggi pada fitur yang relevan dengan domain.

$\sqrt{tf}$: Karena tren menarik yang diamati pada Gambar 3.6, kami memperkenalkan perkiraan korelasi rw/tf menggunakan fungsi akar kuadrat $\sqrt{tf}$, yang menunjukkan perilaku umum kurva yang diplot pada Gambar 3.6.

Untuk setiap skema, kami juga memperkenalkan alternatif rw, dengan mengganti faktor tf di semua skema dengan nilai rw yang dihitung menggunakan model random walk kami. Misalnya, untuk skema $tf.idf$, kami memperkenalkan skema $rw.idf$; untuk $log(tf)$, kami memperkenalkan $log(rw)$; dan seterusnya. Hasil klasifikasi yang diperoleh untuk skema pembobotan yang berbeda pada tiga dataset ditunjukkan pada Tabel 3.8 dan 3.9, dan diilustrasikan secara grafis pada Gambar 3.7 dan 3.8. Uji signifikansi statistik dijalankan untuk membandingkan kinerja alternatif rw dan tf untuk setiap skema pembobotan.

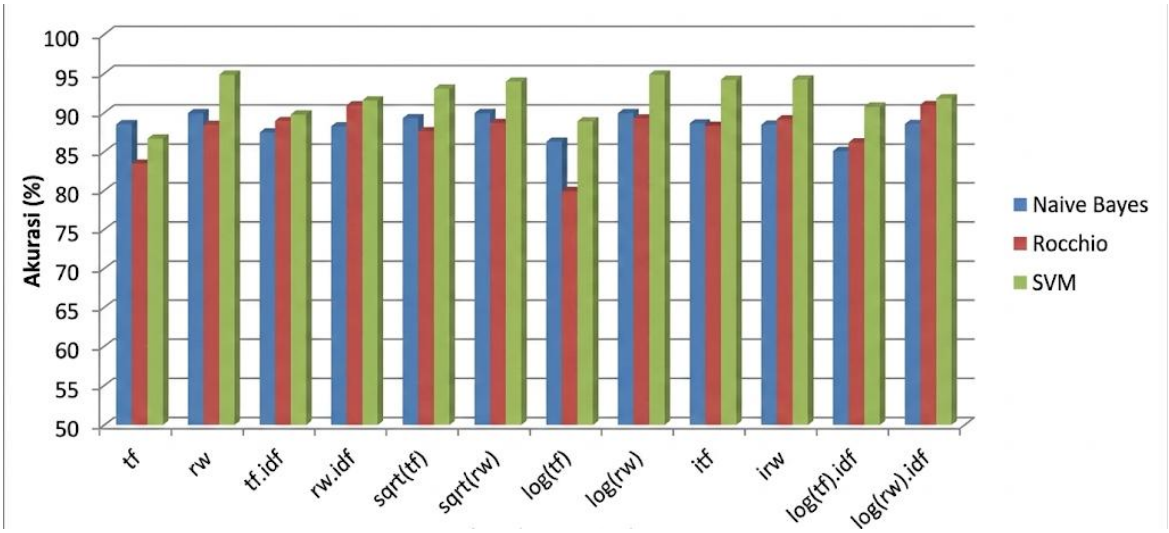
Tabel 3.8 Hasil rata-rata mikro untuk skema pembobotan yang berbeda ($rw^{\leftrightarrow}_{e.oc}$).

Dataset	tf	rw	$tf.idf$	$rw.idf$	$\sqrt{tf}$	$\sqrt{rw}$	$log(tf)$	$log(rw)$	itf	irw	$log(tf).idf$	$log(rw).idf$
Naïve Bayes												
$WebKB_4$	84.2	**86.1***	81.6	**83.8***	85.2	**85.9**	85.1	**85.9**	83.3	83.3	83.2	**83.8**
$WebKB_6$	81.3	**82.4**	78.8	**81.5***	81.5	**81.9**	81.6	**81.9**	77.3	**77.7**	80.4	**81.7**
$LSpam$	99.2	**99.3**	99.1	99.1	99.3	99.3	99.3	**99.4**	99.3	**99.4**	99.0	99.1
$20NG$	89.3	**91.5****	86.9	**88.6****	90.8	**91.4**	77.3	**91.4***	91.2	**91.5**	76.7	**88.4****

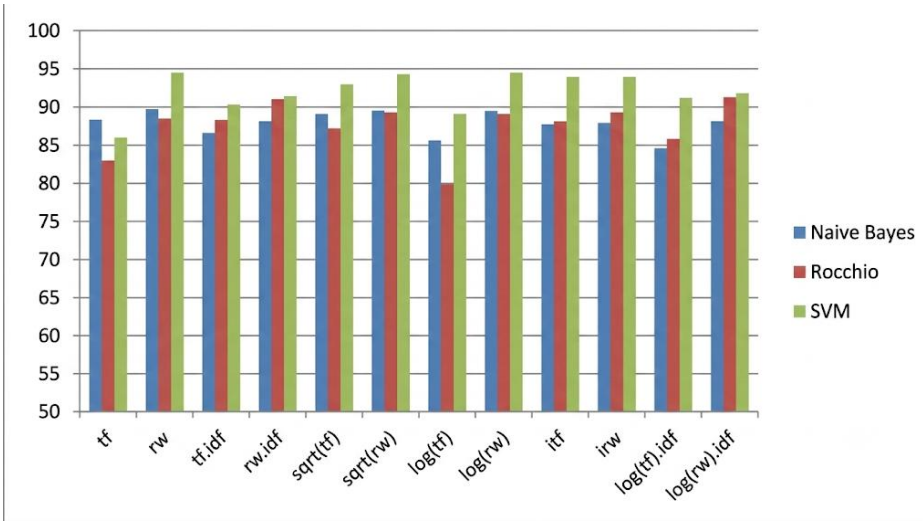
Rocchio												
$WebKB_4$	74.9	**83.5****	84.3	**87.5****	81.2	**84.5****	80.6	**84.2****	82.9	**84.7***	86.5	87.8
$WebKB_6$	70.1	**78.6****	80.1	**84.3****	76.3	**79.9****	75.3	**79.5****	78.1	**80.1***	82.9	**84.3**
LSpam	96.5	**97.8****	98.1	**98.5**	97.4	**98.1**	97.3	**97.9**	97.6	**98.1**	98.2	**98.5**
$20NG$	90.5	**94.8****	91.5	**94.3****	94.1	**94.8***	67.0	**94.9****	94.6	94.7	76.1	**94.4****
SVM												
$WebKB_4$	81.3	**94.1****	88.1	**90.4**	92.0	**93.8***	91.4	**94.0***	92.6	92.9	89.8	90.6
$WebKB_6$	79.3	**90.7****	87.1	**88.7***	89.7	**90.8**	89.9	**90.9***	**89.9**	89.6	88.1	88.6
LSpam	93.6	**98.9****	93.6	**93.9**	98.2	**99.1***	97.6	**99.1****	98.6	98.7	94.5	94.7
$20NG$	90.1	**94.4****	92.5	**93.2***	92.0	**93.9**	77.6	**94.5****	94.6	94.7	93.1	**93.3**

Tabel 3.9 Hasil rata-rata makro untuk berbagai skema pembobotan ($rw^{\leftrightarrow}_{e.oc}$).

Dataset	tf	rw	$tf.idf$	$rw.idf$	$\sqrt{tf}$	$\sqrt{rw}$	$log(tf)$	$log(rw)$	itf	irw	$log(tf).idf$	$log(rw).idf$
Naïve Bayes												
$WebKB_4$	82.5	**83.5***	78.9	**81.2***	82.4	**83.1**	82.4	**83.2**	**78.1**	77.5	80.6	**81.1**
$WebKB_6$	**68.1**	66.5	68.5	**69.8***	**66.3**	65.6	**66.9**	65.7	**52.6**	52.3	69.4	**70.2**
LSpam	98.6	**98.8**	98.4	98.4	98.7	**98.8**	98.8	**98.9**	98.9	98.9	98.3	**98.4**
$20NG$	89.3	**91.4****	86.8	**88.5****	90.7	**91.3**	77.1	**91.3***	91.1	**91.4**	76.5	**88.4****
Rocchio												
$WebKB_4$	74.4	**82.6****	83.4	**86.7****	80.3	**83.5****	79.7	**83.2****	82.0	**83.7***	85.7	86.9
$WebKB_6$	62.7	**71.9****	71.6	**75.8****	69.6	**73.2****	68.6	**72.7****	71.5	**73.1***	74.7	75.9
LSpam	94.1	**96.2****	96.7	**97.3**	95.5	**96.7**	95.5	**96.5**	96.0	**96.7**	97.0	**97.4**
$20NG$	90.5	**94.8****	91.5	**94.3****	94.2	**94.8***	68.1	**94.9****	94.6	**94.8**	76.1	**94.4****
SVM												
$WebKB_4$	77.4	**93.2****	85.8	**88.6****	90.8	**93.0***	90.1	**93.2***	91.7	**91.9**	87.9	**88.8**
$WebKB_6$	68.2	**81.1****	75.9	**77.3***	81.0	**82.0**	80.0	**81.7***	**82.1**	81.3	77.1	77.2
LSpam	86.3	**97.9****	86.1	**86.8**	96.6	**98.3***	95.4	**98.3****	97.6	**97.7**	88.4	**88.8**
$20NG$	90.6	**94.4****	92.6	**93.1***	90.8	**94.0**	77.6	**94.5****	94.6	**94.7**	93.1	**93.3**



Gambar 3.7 Hasil rata-rata mikro keseluruhan untuk berbagai skema pembobotan.



Gambar 3.8 Hasil rata-rata makro keseluruhan untuk berbagai skema pembobotan.

Seperti yang terlihat pada tabel, model random walk kami jelas mengungguli alternatif frekuensi istilah baik dalam rata-rata mikro maupun makro di semua dataset dan semua pengklasifikasi. Dalam kasus terburuk, sistem random walk berkinerja sebaik model dasar frekuensi istilah. Keunggulan model random walk kami menunjukkan bahwa penggunaan dependensi antar fitur dapat menghasilkan peningkatan yang signifikan, dan peningkatan ini konsisten di antara skema pembobotan yang berbeda.

3.6 EVALUASI AKHIR DAN PROSPEK PENGEMBANGAN

Dalam artikel ini, kami memperkenalkan pendekatan random walk untuk pembobotan istilah yang memiliki kemampuan untuk menangkap ketergantungan istilah dalam teks dengan memperhitungkan properti struktural teks. Melalui eksperimen yang dilakukan pada tugas klasifikasi teks, kami menunjukkan bahwa model random walk dapat mencapai pengurangan tingkat kesalahan relatif sebesar 3,2–84,3%, dibandingkan dengan pendekatan berbasis frekuensi istilah tradisional. Hasil evaluasi menunjukkan bahwa kinerja sistem konsisten untuk berbagai ukuran jendela, dan waktu eksekusinya sebanding dengan model $tf.idf$.

Selain itu, dalam eksperimen yang dilakukan menggunakan berbagai kebijakan skema pembobotan, pembobotan istilah random walk secara konsisten ditemukan lebih unggul dibandingkan dengan skema pembobotan frekuensi istilah tradisional. Kami percaya hasil ini mendukung klaim kami bahwa model random walk dapat secara akurat memperkirakan bobot istilah dengan memperhitungkan ketergantungan istilah, dan dapat digunakan sebagai teknik untuk memodelkan distribusi probabilitas fitur dalam sebuah dokumen. Dalam pekerjaan selanjutnya, kami berencana untuk memperluas model ini dan menggunakannya untuk mendefinisikan model bahasa formal, di mana kami dapat memperkirakan probabilitas urutan n-gram kata yang lebih panjang.

BAB 4
PENALARAN PEER-TO-PEER UNTUK ONTOLOGI YANG SALING TERHUBUNG

4.1 PENALARAN TERDISTRIBUSI UNTUK ONTOLOGI WEB TERHUBUNG

Web Semantik umumnya dipahami sebagai web data yang dapat dibaca mesin dan saling terhubung sebagian. Data ini pada dasarnya terdistribusi dan menyerupai struktur web dalam hal sumber daya yang disediakan oleh pihak yang berbeda di lokasi fisik yang berbeda. Sejumlah infrastruktur untuk menyimpan dan menanyakan data web semantik terdistribusi, terutama yang dikodekan dalam RDF, telah dikembangkan. Meskipun ada upaya awal untuk mengintegrasikan penalaran Skema RDF ke dalam pemrosesan kueri terdistribusi, hampir semua pekerjaan pada penalaran logika deskripsi sebagai dasar untuk mengimplementasikan inferensi dalam Bahasa Ontologi Web OWL masih mengasumsikan pendekatan terpusat di mana terminologi lengkap harus ada pada satu sistem dan semua langkah inferensi dilakukan pada sistem ini. Kami telah merancang dan mengimplementasikan metode penalaran terdistribusi yang mempertahankan kebenaran dan kelengkapan penalaran di bawah semantik impor OWL asli dan memiliki sifat-sifat yang menguntungkan terkait komputasi paralel dan overhead yang disebabkan oleh upaya komunikasi dan derivasi tambahan. Metode ini didasarkan pada metode resolusi yang benar dan lengkap untuk logika deskripsi ASC yang kami modifikasi agar berfungsi dalam pengaturan terdistribusi.

Web Semantik umumnya dipahami sebagai web data yang dapat dibaca mesin dan saling terhubung sebagian. Data ini pada dasarnya terdistribusi dan menyerupai struktur web dalam hal sumber daya yang disediakan oleh pihak yang berbeda di lokasi fisik yang berbeda. Tujuan utama teknologi web semantik adalah untuk membuat jaringan data terdistribusi ini dapat diakses dengan cara yang mengabstraksikan distribusi fisik dengan menyediakan alat dan teknologi untuk mengakses dan memproses data terdistribusi seolah-olah disimpan di mesin lokal. Sejumlah infrastruktur untuk menyimpan dan menanyakan data web semantik terdistribusi, terutama yang dikodekan dalam RDF, telah dikembangkan. Meskipun infrastruktur ini mencakup aspek penting dari penanganan data terdistribusi seperti pengindeksan, perutean kueri dan optimasi, penggunaan penalaran logis, landasan penting lain dari teknologi web semantik dalam pengaturan terdistribusi, adalah masalah yang diabaikan oleh sebagian besar pendekatan.

Meskipun ada upaya pertama untuk mengintegrasikan penalaran Skema RDF ke dalam pemrosesan kueri terdistribusi, hampir semua pekerjaan pada penalaran logika deskripsi sebagai dasar untuk mengimplementasikan inferensi dalam Bahasa Ontologi Web OWL masih mengasumsikan pendekatan terpusat di mana terminologi lengkap harus ada pada satu sistem dan semua langkah inferensi dilakukan pada sistem ini. Pendekatan ini memiliki sejumlah kekurangan serius. Pertama-tama, kumpulan data lengkap, yang mungkin sangat besar, harus ditransfer ke sistem penalaran pusat yang menciptakan banyak lalu lintas jaringan. Selain itu,

mentransfer ontologi lengkap ke satu reasoner juga menjadikan ini sebagai hambatan utama dalam sistem. Hal ini dapat mencapai batas data yang dapat diproses oleh sistem penalaran.

Sejumlah pendekatan untuk penalaran terdistribusi tentang ontologi yang saling terkait telah diusulkan yang tidak memerlukan ontologi untuk dikirim ke reasoner pusat. Semua pendekatan ini bergantung pada pembatasan ketat pada jenis tautan antar ontologi atau cara konsep yang didefinisikan dalam ontologi lain dapat digunakan dan disempurnakan, sehingga memperkenalkan jenis tautan khusus antara kumpulan data yang disimpan di lokasi berbeda. Secara khusus, tidak satu pun dari pendekatan ini mendukung definisi standar impor logis dari spesifikasi OWL, sehingga membatasi kegunaannya pada kumpulan data nyata. Kami mengilustrasikan masalah ini menggunakan contoh kecil.

Contoh 4.1 Kita mengasumsikan dua ontologi yang digambarkan pada Gambar 1 yang dihubungkan oleh serangkaian aksioma kesetaraan antara nama konsep dari kedua ontologi tersebut. Perhatikan bahwa aksioma pemetaan juga dapat direpresentasikan dalam bentuk impor OWL. Dalam hal ini, aksioma pemetaan akan menjadi bagian dari salah satu dari dua ontologi dan ontologi lainnya akan diimpor oleh ontologi yang berisi aksioma pemetaan tersebut. Seperti yang dapat kita lihat dengan mudah, ontologi gabungan tersebut tidak konsisten karena ketidaksesuaian dalam definisi konsep 'tuple' yang dalam satu kasus didefinisikan sebagai tidak mengandung himpunan apa pun. Dalam kasus lain, tuple didefinisikan sebagai kasus khusus dari himpunan yang pada gilirannya didefinisikan untuk selalu mengandung setidaknya satu himpunan (himpunan kosong). Konstanta a yang didefinisikan sebagai contoh dari Pair yang pada gilirannya merupakan tuple membuat ontologi tersebut tidak konsisten.

Tujuan kami adalah memiliki metode penalaran terdistribusi yang melakukan penalaran lokal pada dua ontologi dan masih mampu mendeteksi inkonsistensi ontologi gabungan. Melihat proposal sebelumnya untuk penalaran terdistribusi yang disebutkan di atas, kami menyadari bahwa tidak satu pun dari proposal tersebut memenuhi persyaratan ini. Kerangka kerja s-koneksi tidak berlaku dalam skenario ini karena tidak memungkinkan spesifikasi hubungan kesetaraan antara ontologi yang saling terkait. Menggunakan kerangka kerja ekstensi konservatif tidak memberikan keuntungan apa pun dalam hal penalaran lokal. Secara khusus, ontologi gabungan bukanlah ekstensi konservatif dari ontologi A maupun ontologi B karena dalam kedua kasus tersebut, informasi tambahan di bagian lain dapat digunakan untuk memperoleh informasi baru mengenai tanda tangan ontologi A atau ontologi B, misalnya, ketidakpuasan konsep A : *Tuple* dan B : *Tuple*. Pengkodean pemetaan dalam logika deskripsi terdistribusi, pada akhirnya, memungkinkan kita untuk menyimpulkan relasi subsumsi antara konsep A : *Tuple* dan A : *Set*, namun hal ini tidak membuat ontologi gabungan menjadi tidak konsisten sehingga kehilangan beberapa implikasi yang didukung oleh semantik impor asli.

Ontology A	Ontology B
$A{:}Set \sqsubseteq \exists A{:}part.A{:}Set$	$B{:}Tuple \subseteq B{:}Set$
$A{:}Tuple \sqsubseteq \forall A{:}part.\neg A{:}Set$	$B{:}Pair \sqsubseteq B{:}Tuple$
	$B{:}Pair(a)$

Mapping

$$A{:}Tuple \doteq B{:}Tuple$$
$$A{:}Set \doteq B{:}Set$$

Gambar 4.1 Dua ontologi berisi aksioma yang mendeskripsikan konsep Himpunan, Tuple, dan Pasangan, sebuah pencocok ontologi telah membuat pemetaan antara ontologi-ontologi tersebut.

Tujuan kami adalah mengembangkan metode penalaran tentang ontologi logika deskripsi yang mengatasi kekurangan metode yang ada dengan:

- menjaga kebenaran dan kelengkapan penalaran di bawah semantik impor OWL asli
- menghindari pembatasan penggunaan definisi dari ontologi jarak jauh dalam definisi lokal atau pada cara pengetahuan didistribusikan secara a priori
- menghindari (sebagian besar) seluruh ontologi lokal untuk dikirim ke penalaran pusat tetapi mendukung pemrosesan lokal ontologi yang terhubung dan komunikasi terbatas melalui pengiriman pesan
- sifat-sifat yang menguntungkan terkait overhead dalam hal derivasi tambahan, upaya komunikasi yang dibutuhkan, dan penghematan karena eksekusi paralel langkah-langkah inferensi.

Kami telah merancang dan mengimplementasikan metode penalaran terdistribusi yang memenuhi persyaratan ini. Metode ini didasarkan pada metode resolusi yang benar dan lengkap untuk *ASC* logika deskripsi yang kami modifikasi agar berfungsi dalam pengaturan terdistribusi.

Artikel ini disusun sebagai berikut: Pada bagian selanjutnya, kami membahas prinsip-prinsip penalaran terdistribusi sebelum memberikan gambaran umum tentang karya terkait di Bagian 2. Bagian selanjutnya memberikan pengantar singkat tentang penalaran resolusi dan menjelaskan prinsip distribusi yang menjadi dasar pendekatan kami. Bagian 4 menyajikan detail metode penalaran terdistribusi kami, khususnya kami menunjukkan bahwa metode resolusi yang diusulkan dalam untuk logika deskripsi dapat didistribusikan sehingga menghasilkan kalkulus terdistribusi yang valid dan lengkap untuk ontologi ASC. Pada Bagian 5, kami menyelidiki sifat-sifat metode ini sehubungan dengan jumlah derivasi, upaya komunikasi, dan tingkat paralelisasi dan menunjukkan bahwa parameter-parameter ini menjanjikan. Berbagai opsi untuk memperluas implementasi dalam pekerjaan mendatang dibahas pada Bagian 6. Bagian 7 menyimpulkan pekerjaan ini.

Penalaran Terdistribusi

Terdapat berbagai pilihan untuk mendistribusikan proses penalaran logis. Banyak dari pilihan ini telah diteliti di bidang pembuktian teorema otomatis untuk logika orde pertama. Berikut ini kita membahas pilihan-pilihan tersebut dan pro dan kontranya sehubungan dengan persyaratan dan tujuan yang didefinisikan dalam pendahuluan. Secara khusus, kita harus membuat dua pilihan:

(1) Kita harus memilih metode penalaran yang sahih dan lengkap untuk logika deskripsi dan memungkinkan distribusi.
(2) Kita harus memilih prinsip distribusi yang mendukung penalaran lokal dan meminimalkan biaya penalaran dan komunikasi.

Penalaran. Mengenai metode penalaran membedakan antara strategi berbasis pengurutan, pengurangan sub-tujuan, dan berbasis-contoh. Strategi berbasis-contoh adalah implementasi langsung dari metode Herbrand yang menghasilkan contoh dasar teori dan menggunakan metode proposisional untuk menguji kepuasan. Strategi pengurangan sub-tujuan membangun satu bukti pada satu waktu dengan memilih dan menyelesaikan sub-tujuan dan membiarkan model logis tidak berubah. Contoh tipikal strategi pengurangan sub-tujuan adalah metode pemrograman logika dan tabel analitik. Metode berbasis pengurutan, pada akhirnya, didasarkan pada modifikasi terinformasi dari representasi klausa dengan menurunkan klausa baru dan menghapus klausa yang berlebihan hingga klausa kosong diturunkan atau tidak ada kesimpulan baru yang dapat ditarik. Dengan cara ini, metode berbasis pengurutan secara implisit membangun banyak upaya pembuktian secara paralel karena tidak jelas sebelumnya, derivasi mana yang akhirnya berkontribusi pada derivasi klausa kosong. Contoh tipikal metode berbasis pengurutan adalah resolusi dan superposisi dasar.

Tabel analitik merupakan metode dominan untuk mengimplementasikan sistem inferensi yang valid dan lengkap untuk logika deskripsi. Namun, telah ditunjukkan bahwa metode resolusi yang valid dan lengkap untuk logika deskripsi ekspresif dapat didefinisikan. Kami mengecualikan metode lain yang ada seperti reduksi penalaran DL ke pemrograman logika dari investigasi kami karena pendekatan ini tidak valid dan lengkap untuk bahasa yang kami minati. Karena metode berbasis tabel maupun berbasis resolusi memenuhi persyaratan kami terkait cakupan bahasa dan kelengkapan, faktor penentunya adalah kesesuaiannya untuk penalaran terdistribusi.

Prinsip Distribusi. Survei membahas berbagai strategi untuk memparalelkan inferensi logis. Secara khusus, penulis membedakan antara paralelisme pada tingkat istilah, klausa, dan pencarian. Gagasan paralelisme pada tingkat istilah dan klausa adalah untuk mempercepat fungsi penalaran dasar seperti pencocokan, penyatuan, atau langkah resolusi tunggal dengan mengeksekusinya secara paralel menggunakan memori bersama untuk meningkatkan kinerja. Jelas, pendekatan ini tidak sesuai untuk tujuan kita karena tidak membayangkan distribusi aksioma ontologi tetapi hanya bertujuan pada eksekusi paralel metode penalaran dasar. Paralelisme pada tingkat pencarian berarti eksekusi paralel dari keseluruhan proses derivasi dan dapat dibedakan lebih lanjut menjadi pendekatan pencarian multi-arah dan pencarian terdistribusi.

Sementara dalam kasus pertama, beberapa proses pencarian, seringkali dengan heuristik yang berbeda atau titik awal yang berbeda dijalankan secara paralel, pendekatan pencarian terdistribusi menetapkan bagian dari ruang pencarian ke masing-masing reasoner dan mengoordinasikan keseluruhan proses pencarian dengan bertukar hasil sementara. Sementara pendekatan pertama membutuhkan ontologi lengkap untuk tersedia di semua reasoner, paradigma pencarian terdistribusi secara alami sesuai dengan penyimpanan terdistribusi bagian-bagian ontologi dan oleh karena itu mewakili paradigma yang sesuai dengan tujuan pembelajaran kita karena memungkinkan untuk menetapkan bagian dari ruang pencarian yang relevan untuk ontologi tertentu ke instance reasoner lokal yang berinteraksi dengan reasoner lokal lainnya jika perlu.

Pilihan paradigma pencarian terdistribusi memiliki konsekuensi terhadap pilihan metode penalaran. Secara khusus, telah ditunjukkan bahwa pencarian terdistribusi dapat digunakan dalam kombinasi dengan metode berbasis pengurutan untuk mendukung eksekusi paralel penalaran logis. Kami membangun di atas hasil ini dengan mengusulkan metode penalaran terdistribusi berdasarkan prinsip resolusi. Proposal kami melampaui keadaan seni dalam pembuktian teorema terdistribusi karena membahas subset spesifik yang dapat diputuskan dari logika orde pertama yang belum diteliti dalam konteks pembuktian teorema terdistribusi. Lebih lanjut, strategi yang ada untuk menetapkan langkah inferensi ke penalaran seperti kriteria grafik leluhur tidak dapat menghindari redundansi. Kami menunjukkan bahwa penetapan klausa ke penalaran berdasarkan lokasi fisik dari aksioma yang sesuai menghindari redundansi untuk kasus khusus resolusi terurut yang dapat digunakan sebagai dasar untuk metode penalaran terdistribusi yang sahih dan lengkap untuk logika ALC.

4.2 KONTEKS DAN REFERENSI PENGEMBANGAN

Pendekatan yang mengimplementasikan penalaran terdistribusi pada ontologi logika deskripsi relatif jarang. Distribusi ontologi RDFS lebih populer karena mengadaptasi mekanisme inferensi berbasis aturan dari ontologi RDFS ke pengaturan terdistribusi relatif mudah dibandingkan dengan mendistribusikan metode tableau yang biasanya digunakan untuk logika deskripsi. Pendekatan kami lebih terkait dengan beberapa metode terdistribusi untuk penalaran orde pertama karena mereka menggunakan penalaran resolusi.

Penalaran RDF Terdistribusi

Marvin adalah platform untuk pemrosesan paralel dan terdistribusi data RDF pada jaringan rekan yang terhubung secara longgar. Metode penalaran diilustrasikan pada komputasi penutupan deduktif. Meskipun tidak ada kontradiksi dan bukti dalam RDF(S), proses komputasi penutupan deduktif sebanding dengan saturasi himpunan klausa. Oleh karena itu, metode yang diimplementasikan oleh Marvin dapat diklasifikasikan sebagai strategi pencarian terdistribusi.

Penalaran DL Modular

Pendekatan saat ini untuk penalaran terdistribusi pada logika deskripsi sebagian besar bergantung pada metode tableau. Distribusi dengan menyelesaikan dua pilihan aturan tabel nondeterministik secara paralel sulit dilakukan karena menghambat penerapan strategi

optimasi dan pemblokiran. Sebaliknya, sebagian besar pendekatan tabel terdistribusi mencoba mengidentifikasi semua kemungkinan konflik, yaitu semua aksioma yang mungkin berasal dari modul lain dan akan menyebabkan kontradiksi, dan mengirimkannya sebagai kueri ke modul lain. Sejauh ini, hal ini hanya dilakukan untuk tautan dengan ekspresivitas yang agak terbatas antara modul. Implementasi penalaran T-Box terdistribusi yang paling menonjol untuk ontologi adalah Logika Deskripsi Terdistribusi (DDL), sebuah strategi pencarian terdistribusi. Ini hanya mendukung jenis tautan khusus (aturan jembatan) antara ontologi. Domain lokal harus terpisah, yaitu tidak ada subsumsi nyata antara elemen dari modul yang berbeda.

Seperti DDL, $\varepsilon - koneksi$ memperlakukan domain lokal sebagai terpisah dan tidak mendukung relasi subsumsi antara modul. $\varepsilon - koneksi$ dapat digunakan untuk menghubungkan ontologi dengan ekspresivitas yang berbeda dan jaringan yang dihasilkan dapat diterjemahkan ke logika deskripsi umum. Pelet penalaran tableau telah diperluas untuk mendukung $\varepsilon - koneksi$ tetapi penalaran dilakukan oleh satu penalaran pelet dan tidak terdistribusi. mengusulkan pendekatan untuk ontologi modular berdasarkan ekstensi konservatif. Ini bukan metode penalaran terdistribusi tetapi strategi untuk membuat jaringan ontologi dengan ontologi mandiri sehingga penalaran pada ontologi secara terpisah menjadi lengkap. Secara kasar, ontologi A adalah ekstensi konservatif dari ontologi B lainnya, jika tidak ada aksioma yang tersirat oleh A dan tidak tersirat oleh B yang hanya menggunakan simbol dari tanda tangan lokal B. Oleh karena itu, untuk penalaran dalam ontologi gabungan tentang simbol B, ontologi A dapat diabaikan dengan aman, karena tidak berpengaruh pada semantik B. Ide dari pendekatan ini adalah, penalaran sangat sederhana, ketika jaringan ontologi merupakan ekstensi konservatif dari masing-masing ontologi.

Meskipun penalaran aktual relatif mudah dalam pendekatan ini, tugas yang kompleks adalah membuat ontologi yang sesuai dengan persyaratan ekstensi konservatif yang kuat. Metode untuk mengekstraksi jenis modul ontologi mandiri dari ontologi yang lebih besar diusulkan. mengusulkan algoritma tableau terdistribusi untuk Logika Deskripsi Berbasis Paket (PDL). Bukti kebenaran dan kelengkapan algoritma diberikan tetapi metode tersebut tidak diimplementasikan dan tidak dievaluasi. Secara khusus, tidak ada bukti yang diberikan bahwa komputasi terdistribusi dari algoritma tersebut sebenarnya lebih cepat daripada komputasi pada satu mesin. KAONp2p adalah infrastruktur untuk menjawab kueri atas ontologi terdistribusi berdasarkan sistem KAON2. Pendekatan ini berfokus pada manajemen ontologi dan pemilihan pengetahuan untuk penalaran tentang pernyataan (Aboxes). Bagian-bagian yang relevan dari terminologi disalin ke rekan yang menjawab kueri.

Metode resolusi terdistribusi

Resolusi adalah metode yang paling populer untuk pembuktian FOL. Pendekatan untuk penalaran orde pertama terdistribusi dimotivasi oleh pertimbangan efisiensi, kinerja ditingkatkan dengan menggunakan beberapa prosesor secara paralel, misalnya, adalah paralelisasi dari penalaran orde pertama Otter yang banyak digunakan (kembali). Sementara pembukti resolusi umum memilih satu klausa yang disebut klausa yang diberikan dan menyelesaikannya dengan semua klausa pasangan yang mungkin, Roo memilih beberapa

klausa yang diberikan secara paralel dan menyelesaikan konflik yang muncul dari proses paralel. Berbeda dengan metode terdistribusi yang disebutkan sejauh ini, Roo menggunakan memori bersama dan dapat diklasifikasikan sebagai paralelisasi pada tingkat klausa menurut.

Penalaran Berbasis Partisi adalah strategi pencarian terdistribusi yang membutuhkan penalaran lokal yang lengkap untuk menemukan konsekuensi. Persyaratan kelengkapan yang kuat untuk penalaran lokal pasti menyebabkan derivasi lebih banyak klausa daripada yang diperlukan untuk penyangkalan. Meskipun demikian, metode distribusi telah terbukti mempercepat beberapa strategi resolusi dalam pengaturan paralel dengan memori bersama. Perhatikan bahwa kalkulus resolusi yang diterapkan untuk metode penalaran terdistribusi kami tidak lengkap untuk menemukan konsekuensi, untuk pertimbangan efisiensi, kalkulus tersebut dirancang untuk menghasilkan sejumlah klausa yang relatif kecil. mengusulkan metode penalaran terdistribusi untuk teori proposisional. Para penulis secara otomatis membuat serangkaian teori klausa yang dihubungkan oleh variabel bersama dan menyelidiki korelasi antara karakteristik jaringan teori yang dibuat dan kueri (misalnya, jumlah dan panjang klausa penghubung) dan karakteristik pemrosesan kueri (misalnya, kedalaman kueri). Waktu eksekusi kueri diselidiki untuk kompleksitas jaringan teori dan kueri yang berbeda, tetapi tidak dibandingkan dengan komputasi terpusat.

4.3 MENDISTRIBUSIKAN RESOLUSI LOGIS

Sebelum menjelaskan metode resolusi terdistribusi kami untuk ontologi, kami terlebih dahulu secara singkat meninjau penalaran resolusi standar dan menyajikan ide dasar untuk mendistribusikan resolusi sesuai dengan paradigma pencarian terdistribusi.

Pembuktian teorema resolusi

Resolusi adalah metode penalaran yang sangat populer untuk pembukti logika orde pertama (FOL). Karena logika deskripsi merupakan himpunan bagian yang ketat dari logika orde pertama, resolusi juga dapat diterapkan pada ontologi. Untuk tujuan ini, ontologi DL diubah menjadi himpunan klausa orde pertama seperti yang didefinisikan dalam Bagian 4.1. Gambar 4.2 mengilustrasikan terjemahan ini pada contoh dari Gambar 4.1. Terjemahan ini dapat dilakukan berdasarkan aksioma secara independen dari bagian lain dari ontologi. Dapat ditunjukkan bahwa ontologi konsisten jika dan hanya jika himpunan klausa konsisten. Himpunan klausa konsisten jika dan hanya jika penerapan menyeluruh dari aturan resolusi standar tidak menghasilkan klausa kosong. Resolusi standar terdiri dari dua aturan, aturan resolusi utama disertai dengan aturan pemfaktoran.

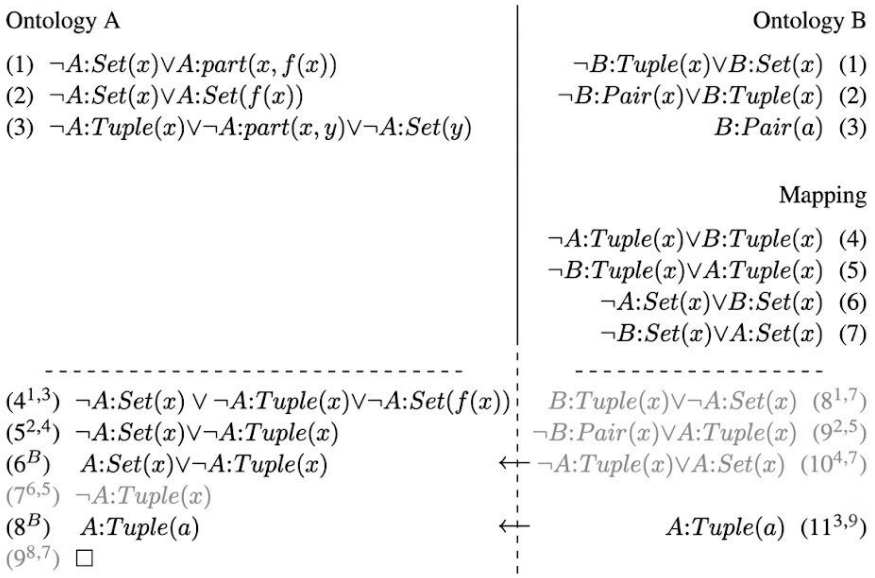
Definisi 1. (Resolusi Standar) Untuk klausa C dan D dan literal A dan $\neg B$, resolusi standar didefinisikan oleh aturan

$$\text{Resolution } \frac{C \vee A \vee D \vee \neg B}{C\sigma \vee D\sigma}$$

$$\text{Factoring } \frac{C \vee A \vee B}{C\sigma \vee A\sigma}$$

di mana substitusi σ adalah penyatu paling umum dari A dan B. Dalam kebanyakan kasus, aturan resolusi saja menghasilkan klausa kosong jika himpunan klausa masukan tidak konsisten. Tetapi, untuk mendapatkan klausa kosong misalnya dari dua klausa $P(x) \cap P(y)$ dan $\neg P(a) \cap \neg P(b)$, pemfaktoran diperlukan. Dimungkinkan juga untuk menggabungkan kedua aturan tersebut menjadi satu aturan inferensi.

Resolusi dengan pemfaktoran sudah lengkap untuk memutuskan kepuasan teori orde pertama, tetapi untuk penalaran resolusi yang efisien, aturan reduksi harus ditambahkan yang menghapus klausa yang berlebihan. Aturan reduksi mirip dengan aturan inferensi, perbedaannya adalah premis dihapus saat menerapkan aturan tersebut. Aturan reduksi yang paling penting adalah penghapusan subsumsi, aturan ini menghapus klausa yang disubsumsi oleh klausa lain, misalnya klausa $P(x)\ V\ Q(x)\ V\ R(x)$ dan $P(x)\ V\ Q(x)$ digantikan oleh $P(x)\ V\ Q(x)$ (yaitu klausa yang disubsumsi $P(x) V\ Q(x)\ V\ R(x)$ dihapus).



Gambar 4.2 Contoh resolusi terurut terdistribusi pada ontologi dari Gambar 1. Aksioma diterjemahkan ke klausa yang ditentukan oleh daftar literal, semua variabel dikuantifikasi secara universal secara implisit. Kuantifikasi eksistensial diterjemahkan ke fungsi skolem. Klausa yang diinferensikan digambarkan di bawah garis putus-putus, angka superskrip mengacu pada asal klausa, yaitu premis yang diturunkan darinya atau ontologi yang mengirimkannya. Panah menunjukkan propagasi klausa ke ontologi lain. Prioritas simbol adalah $B : Set > B : Tuple > B : Pair > A : Set > A : Tuple > A : part$, semua literal dengan simbol teratas A : part dipilih. Untuk membantu melacak inferensi, literal yang dapat diselesaikan diatur dalam warna hitam, literal lainnya dalam warna abu-abu. Oleh karena itu, klausa turunan terdiri dari literal abu-abu dari premisnya.

Definisi 2. (Reduksi Subsumsi) Untuk klausa C dan D, reduksi subsumsi didefinisikan oleh aturan

$$\text{Subsumption Reduction } \frac{C\ \ D}{C}$$

di mana literal dari $C\sigma$ merupakan subset dari literal dari D untuk beberapa substitusi σ.

Resolusi Terdistribusi

Algoritma 1 mengontrol penerapan aturan resolusi pada sekumpulan klausa masukan. Untuk mencatat klausa mana yang telah diselesaikan satu sama lain, himpunan klausa dibagi menjadi dua himpunan, himpunan klausa yang dapat digunakan (Us) yang harus diselesaikan dan himpunan yang telah dikerjakan (Wo). Klausa dalam himpunan Wo jenuh, yaitu setiap klausa yang dapat diturunkan dari Wo sudah terkandung dalam Wo atau Us atau redundan. Hingga seluruh himpunan klausa jenuh atau ditemukan klausa kosong, algoritma berulang kali memilih klausa (klausa yang diberikan) dari himpunan Us dan memindahkannya ke himpunan Wo (baris 4 hingga 6).

Kemudian resolusi diterapkan pada klausa yang diberikan dan setiap klausa dari himpunan Wo yang dapat diselesaikan dengan yang diberikan (misalnya, berisi literal yang dapat disatukan dengan literal negasi dari klausa yang diberikan) untuk mendapatkan klausa baru. Reduksi dilakukan pada baris 8, fungsi "reduksi" mencakup penghapusan klausa redundan di semua himpunan klausa. Setelah reduksi, klausa baru yang tersisa ditambahkan ke himpunan Us. Reduksi merupakan bagian penting dari pembukti resolusi dan komponen yang paling memakan waktu. Tanpa reduksi, jumlah klausa dalam himpunan Us meningkat jauh lebih cepat daripada klausa yang dipindahkan ke Wo, oleh karena itu Us tidak pernah kosong dan tidak mungkin untuk memenuhi bahkan himpunan klausa kecil dengan konsumsi waktu dan penyimpanan yang wajar. Struktur aturan reduksi sama dengan aturan inferensi, tetapi premis dihapus pada aplikasi. Algoritma 1 adalah versi sederhana dari pembukti resolusi yang ditentukan.

Algorithm 1 Resolution Prover

IsSatisfiable$(KB)[t]$

1: $Wo \leftarrow \emptyset$
2: $Us \leftarrow KB$
3: **while** $Us \neq \emptyset \square \notin$ **and** Us **do**
4: $Given \leftarrow$ CHOOSE(Us)
5: $Us \leftarrow Us \setminus \{Given\}$
6: $Wo \leftarrow Wo \cup \{Given\}$
7: $New \leftarrow$ RESOLVE(Given, Wo)
8: $(Given, Us, New) \leftarrow$ REDUCE$(Given, Us, New)$
9: $Us \leftarrow Us \cup New$
10: **end while**
11: **if** $\square \in$ Us **then return return false**
12: **else return return true**
13: **end if**

Algoritma dasar ini dapat dimodifikasi untuk mendukung penalaran terdistribusi. Secara khusus, inferensi dapat didistribusikan ke berbagai reasoner dengan memisahkan himpunan

klausa input dan menjalankan pembuktian pada bagian-bagian terpisah dari himpunan tersebut:

- Setiap reasoner secara terpisah memenuhi himpunan klausa yang ditugaskan kepadanya.
- Klausa yang baru diturunkan disebarkan ke reasoner lain jika perlu.

Penyebaran klausa ditambahkan ke baris 9 Algoritma 1: alih-alih langsung menambahkan klausa baru ke himpunan Us lokal, beberapa klausa baru dapat ditambahkan ke himpunan Us reasoner lain dan klausa baru yang diterima dari reasoner lain dapat ditambahkan ke himpunan Us lokal.

Definisi 3. (Alokasi) Alokasi untuk himpunan C klausa dan himpunan modul M adalah relasi $a \in (C \times M)$ sedemikian sehingga

$$\forall c \in C : \exists m \in M : a(c, m).$$

Himpunan modul yang dialokasikan ke klausa c oleh alokasi a didefinisikan oleh

$$a(c) := \{m \in M \mid a(c, m)\}.$$

Jika relasi alokasi bersifat fungsional, kita dapat menghilangkan tanda kurung dan menulis $a(c) = m$. Selain propagasi klausa, kita harus menambahkan modifikasi kedua pada algoritma untuk mengubahnya menjadi algoritma resolusi terdistribusi. Berbeda dengan kasus terpusat, sebuah reasoner yang telah memenuhi himpunan klausa lokal mungkin harus melanjutkan penalaran setelah klausa baru diterima dari reasoner lain. Seluruh sistem reasoner yang terhubung berhenti jika klausa kosong diturunkan oleh salah satu reasoner atau semuanya telah jenuh. Setelah deskripsi intuitif tentang algoritma resolusi terdistribusi ini, kita mendefinisikan resolusi terdistribusi secara formal: ‘

Definisi 4. (Kalkulus Resolusi Terdistribusi) Kalkulus resolusi terdistribusi $R(a)$ adalah kalkulus resolusi yang bergantung pada relasi alokasi $a : C \rightarrow M$ sedemikian sehingga setiap aturan r dari $R(a)$ dibatasi pada premis $P \subset C$ dengan

$$\exists m \in M : \forall c \in P : a(c, m).$$

Oleh karena itu, aturan kalkulus resolusi terdistribusi dibatasi pada premis yang dialokasikan ke modul yang sama. Kalkulus terdistribusi dapat diperoleh dari kalkulus resolusi apa pun dengan mendefinisikan relasi alokasi dan menambahkan pembatasan alokasi ke setiap aturan kalkulus.

Definisi 5. (Pembatasan Alokasi) Diberikan aturan r dari kalkulus resolusi dan fungsi alokasi a untuk himpunan klausa C dengan rentang M, pembatasan alokasi adalah

$$\exists m \in M : \exists c \in P : a(c, m)$$

di mana P adalah himpunan premis dari aturan r.

Jelas, penghentian kalkulus yang mendasarinya dipertahankan oleh distribusi jika tidak bergantung pada aturan reduksi. Dalam kasus terburuk, setiap inferensi dari kalkulus asli dilakukan sekali di setiap modul kalkulus terdistribusi. Hasil yang disajikan di Bagian 5 juga menunjukkan bahwa reduksi lokal (yaitu menghapus klausa yang redundan sehubungan dengan reasoner yang memprosesnya) cukup dalam praktiknya.

Mempertahankan kelengkapan tanpa mengalokasikan setiap klausa ke setiap reasoner lebih sulit, kita harus memastikan pembatasan alokasi tidak pernah mengecualikan inferensi yang mungkin dalam kalkulus asli. Merujuk pada Algoritma 1, kita harus memastikan klausa tertentu diselesaikan dengan setiap klausa yang cocok di salah satu himpunan Wo terdistribusi. Oleh karena itu, untuk resolusi standar, klausa C tertentu harus disebarkan ke reasoner mana pun yang himpunan Wo-nya berisi klausa dengan literal yang cocok (yaitu, dapat disatukan dan memiliki polaritas berlawanan) dengan literal apa pun di C. Ini akan menyebabkan overhead komunikasi yang substansial dan berpotensi langkah inferensi yang berlebihan. Untuk menghindari redundansi, kita bertujuan untuk mengalokasikan setiap klausa hanya ke satu reasoner. Alokasi fungsional menjamin bahwa langkah resolusi yang sama tidak pernah dilakukan dua kali, karena klausa yang setara selalu ditugaskan ke reasoner unik yang sama yang menangani penghindaran redundansi lokal. Di bagian selanjutnya, kita akan menunjukkan bagaimana hal ini dapat dicapai tanpa mengorbankan kelengkapan.

4.4 RESOLUSI TERDISTRIBUSI UNTUK ALC

Seperti yang telah kita lihat di atas, kemampuan untuk mendefinisikan metode penalaran terdistribusi yang valid dan lengkap bergantung pada dua persyaratan: (1) keberadaan kalkulus resolusi yang valid dan lengkap dan (2) kemampuan untuk menemukan fungsi alokasi yang sesuai yang memenuhi batasan alokasi. Pada bagian ini, penulis menunjukkan bahwa untuk kasus ontologi yang didefinisikan dalam ALC, kedua persyaratan ini dapat dipenuhi sehingga menghasilkan metode resolusi terdistribusi yang valid dan lengkap. Penulis tidak membahas aturan reduksi pada bagian ini karena reduksi tidak diperlukan untuk menjamin sifat teoritis dari kalkulus yang diusulkan. Namun, untuk penalaran yang efisien, reduksi sangat penting dan oleh karena itu efek praktis distribusi pada reduksi dibahas pada bagian eksperimental.

Sebelum menyajikan kalkulus yang menjadi dasar metode terdistribusi penulis, penulis mendefinisikan logika deskripsi yang penulis gunakan dan terjemahan aksioma logika deskripsi ke klausa orde pertama.

Tabel 4.1 Sintak $\boldsymbol{ALC}$

$\boldsymbol{\alpha ::= C \sqsubseteq C \mid C \equiv C \mid C(x) \mid R(x,x)}$
$\boldsymbol{C ::= \mathbf{T} \mid \bot \mid A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists R.C}$
$\boldsymbol{A ::= namakonsep}$
$\boldsymbol{R ::= namaproperti}$
$\boldsymbol{x ::= namaindividual}$

Logika Deskripsi ALC**.** Ontologi ALC adalah himpunan O dari aksioma α yang dibangun dari konsep C sesuai dengan sintaks berikut yang diberikan dalam Bentuk Backus-Naur pada Tabel 4.1.

Tanda tangan ontologi$Sig(O)$ adalah gabungan terpisah dari nama konsep, nama properti, dan nama individu. Semantik ALC didefinisikan oleh pemetaan konsep dan aksioma ke logika orde pertama. Misalnya, aksioma $ASCSet \sqsubseteq \exists part.Set$ diterjemahkan menjadi $\forall x: Set(x) \rightarrow \exists y: part(x,y) \wedge Set(y)$. Penerjemahan ontologi logika deskripsi ke FOL adalah konjungsi dari aksioma yang diterjemahkan. Perhatikan bahwa nama konsep sesuai dengan predikat uner, nama properti sesuai dengan predikat biner. Normalisasi yang disajikan pada subbagian berikutnya sangat menyederhanakan penerjemahan ke klausa orde pertama, oleh karena itu penulis memberikan pemetaan yang disederhanakan setelah menjelaskan normalisasi.

Normalisasi. Kalkulus resolusi yang penulis terapkan membutuhkan klausa orde pertama sebagai masukan, oleh karena itu rumus orde pertama yang diperoleh dari ontologi diterjemahkan ke dalam klausa. Untuk menjamin penghentian kalkulus resolusi yang diterapkan, ontologi harus dinormalisasi sebelum klausaifikasi. Ini memastikan bahwa hanya jenis aksioma dan klausa yang sesuai yang muncul dalam prosedur penalaran. Untuk kesederhanaan, penulis mengasumsikan ontologi hanya berisi aksioma subsumsi $A \sqsubseteq C$ di mana A bukan konsep kompleks dan tidak ada aksioma ekuivalensi. Subsumsi kompleks $C \pm D$ setara dengan $\mathsf{T} \sqsubseteq \neg C \sqcup D$ dan ekuivalensi $C \equiv D$ dapat digantikan oleh dua subsumsi $C \sqsubseteq D$ dan $D \sqsubseteq C$. Normalisasi bentuk definitorial yang penulis gunakan menggantikan konsep kompleks C di sisi kanan aksioma dengan nama konsep baru A dan menambahkan aksioma $A \sqsubseteq C$ ke ontologi. Dengan demikian, ia memecah aksioma bersarang menjadi aksioma sederhana dengan memperkenalkan konsep baru. Misalnya, Aksioma $A \sqsubseteq B \sqcup \exists r.C$ digantikan oleh aksioma $A \sqsubseteq B \sqcup Q$ dan $Q \sqsubseteq \exists r.C$.

Definisi 6. (Bentuk Definitorial) Untuk subsumsi sederhana $A \sqsubseteq D$ dengan konsep atomik A, Bentuk Definitorial didefinisikan sebagai berikut:

$$\mathrm{Def}(A \sqsubseteq D) := \begin{cases} \{A \sqsubseteq D\}, & \textit{jika semua subistilah dari D adalah konsep literal} \\ \{Q \sqsubseteq D|_p\} \cup \mathrm{Def}(A \sqsubseteq D[D]_p), & \textit{jika } D|_p \textit{ bukan konsep literal} \end{cases}$$

di mana konsep literal adalah nama konsep atau nama konsep yang dinegasikan dan Q adalah nama konsep baru. $D|_p$ menentukan subistilah dari D dan $D[Q]_p$ diperoleh dari D dengan mengganti subistilah ini dengan Q.

Klausifikasi. Setelah normalisasi, ontologi hanya berisi aksioma sederhana yang diterjemahkan ke klausa orde pertama sebagai berikut:

$A \sqsubseteq B$	$\neg A(x) \vee B(x)$
$A \sqsubseteq B \sqcap C$	$\neg A(x) \vee B(x)$
	$\neg A(x) \vee C(x)$
$A \sqsubseteq B \sqcup C$	$\neg A(x) \vee B(x) \vee C(x)$

$$\begin{array}{ll} A \sqsubseteq \exists r.B & \neg A(x) \vee r\big(x, f(x)\big) \\ & \neg A(x) \vee B\big(f(x)\big) \\ A \sqsubseteq \forall r.B & \neg A(x) \vee \neg r(x,y) \vee B(y) \end{array}$$

Misalnya, aksioma $A \sqsubseteq B$ sesuai dengan rumus orde pertama $\forall x: A(x) \rightarrow B(x)$ yang setara dengan klausa $\neg A(x) \vee B(x)$. Seperti biasa, semua variabel secara implisit dikuantifikasi $\forall$-kali, kuantifikasi eksistensial diterjemahkan menggunakan fungsi skolem. Klausa yang dihasilkan dari ontologi pada Gambar 4.1 digambarkan pada Gambar 4.2.

Metode resolusi lengkap untuk ALC

Telah ditunjukkan bahwa metode resolusi standar dapat diadaptasi untuk memberikan penalaran yang tepat dan lengkap untuk ALC. Metode-metode yang menjadi dasar pekerjaan penulis ini bergantung pada resolusi terurut, yang bergantung pada dua parameter: Urutan literal dan fungsi seleksi yang memetakan setiap klausa ke subset literal negatifnya. Parameter-parameter ini digunakan untuk membatasi penerapan aturan inferensi, sehingga mengurangi jumlah klausa turunan dan menghindari inferensi yang berlebihan. Kalkulus resolusi terurut terdiri dari dua skema inferensi berikut: resolusi terurut dan pemfaktoran. Huruf C dan D mengacu pada klausa, A dan B adalah literal.

Definisi 7. (Resolusi Terurut)

$$\text{Resolusi Terurut} = \frac{C \vee A \quad D \vee \neg B}{C\sigma \vee D\sigma}$$

di mana

(1) σ adalah penyatu paling umum dari A dan B
(2) $\neg B$ dipilih dalam $D \vee \neg B$ atau tidak ada yang dipilih dalam $D \vee \neg B$ dan $B\sigma$ maksimal terhadap $D\sigma$
(3) $A\sigma$ benar-benar maksimal terhadap $C\sigma$
(4) tidak ada yang dipilih dalam $C\sigma \vee A\sigma$

$$\text{Pemfaktoran Positif} = \frac{C \vee A \vee B}{C\sigma \vee A\sigma}$$

di mana

(1) σ adalah penyatu paling umum dari A dan B
(2) $A\sigma$ maksimal terhadap $C\sigma \vee B\sigma$
(3) tidak ada yang dipilih dalam $C\sigma \vee A\sigma \vee B\sigma$.

Kalkulus ini dikenal sebagai kalkulus yang sahih dan lengkap untuk klausa orde pertama. Namun, untuk menjamin terminasi pada ALC, diperlukan parameterisasi khusus. Secara khusus, menunjukkan bahwa dengan menggunakan resolusi terurut, keterputusan pada ALC dapat dipertahankan dengan mengubah ontologi menjadi bentuk definitorial sebelum klausifikasi dan menerapkan resolusi terurut dengan seleksi dan pengurutan yang sesuai: Jika

klausifikasi standar diterapkan pada ontologi ALC dalam bentuk definitorial, kepuasan klausa yang dihasilkan dapat diputuskan dengan kalkulus berikut:

Definisi 8. (Resolusi ALC) $Resolusi\ ALC\ R_{ALC}$ adalah kalkulus

- yang terdiri dari aturan inferensi resolusi terurut dan pemfaktoran,
- dengan seleksi literal biner negatif yang tepat, dan
- pengurutan literal $>$ dengan $R\big(x, f(x)\big) > \neg C(x)$ dan $D\big(f(x)\big) > \neg C(x)$, untuk semua simbol fungsi f, dan predikat R, C, dan D.

Persyaratan pengurutan dipenuhi oleh setiap pengurutan jalur leksikografis (LPO, Definisi 9) berdasarkan prioritas total $>$ pada simbol fungsi, predikat, dan logika dengan $f > P > \neg$ untuk setiap simbol fungsi f dan simbol predikat P.

Definisi 9. (Pengurutan Jalur Leksikografis) Pengurutan jalur leksikografis (LPO) adalah pengurutan istilah yang diinduksi oleh prioritas ketat yang beralasan $>$ atas simbol fungsi, predikat, dan logika, yang didefinisikan oleh:

$s = f(s_1, \ldots, s_m) > g(t_1, \ldots, t_n) = t\ jika\ dan\ hanya\ jika$
$t\ adalah\ subistilah\ yang\ tepat\ dari\ s\ atau\ setidaknya\ salah\ satu\ dari\ berikut\ ini\ berlaku$

(i) $f > g$ dan $s > t_i$ untuk semua i dengan $1 \leq i \leq n$
(ii) $f = g$ dan untuk beberapa j kita memiliki $(s_1, \ldots, s_{j-1}) = (t_1, \ldots, t_{j-1}), s_j > t_j$ dan $s > t_k$ untuk semua k dengan $j < k \leq n$
(iii) $s_j < t$ untuk beberapa j dengan $1 \leq j \leq m$.

LPO memiliki properti subterm, yaitu $t > t'$ untuk semua term t' yang merupakan subterm dari term t. Lebih lanjut, jika $>$ bersifat total, LPO yang diinduksi oleh $>$ bersifat total pada term dasar. Untuk tipe literal sederhana yang muncul dalam klausa yang diterjemahkan dari aksioma ASC (Tabel 4.2 yang dijelaskan di bawah), pengurutan LPO dengan prioritas seperti yang dipersyaratkan oleh Definisi 8 dipecah menjadi definisi pengurutan sederhana. Literal yang berisi variabel berbeda tidak dapat dibandingkan, untuk literal yang memiliki variabel yang sama, pengurutannya ditentukan oleh tiga aturan:

- Literal yang berisi simbol fungsi mendahului literal yang tidak berisi simbol fungsi.
- Literal yang berisi simbol fungsi diurutkan sesuai dengan prioritas simbol fungsi.
- Literal yang tidak mengandung simbol fungsi diurutkan menurut prioritas simbol predikat.

Tabel 4.2 Tipe klausa yang dihasilkan dari penerjemahan ontologi ASC ke klausa orde pertama.

	Tipe klausa $\boldsymbol{ALC}$ Literal	yang dapat diselesaikan
1	$R(x, f(x)) \vee \mathbf{P}(x)$	$R(x, f(x))$
2(*a*)	$\mathbf{P}(x)$	$(\neg)P(x)$
2(*b*)	$\mathbf{P_1}\big(\mathbf{f}(x)\big) \vee \mathbf{P_2}(x)$	$(\neg)P(f(x))$
3	$\neg R(x, y) \vee \mathbf{P_1}(x) \vee \mathbf{P_2}(y)$	$\neg R(x, y)$
4	$\mathbf{P}(a)$	$(\neg)P(a)$

5	$(\neg)R(a,b)$	$(\neg)R(a,b)$

Penulis secara singkat meninjau bukti penghentian R_{ALC} pada ontologi ALC karena relevan untuk distribusi kalkulus.

Teorema 1 (Penghentian R_{ALC}). Untuk basis pengetahuan $ALC\ KB$, penjenuhan klausa yang diperoleh dari bentuk definitorial KB oleh R_{ALC} menentukan kepuasan KB dan berjalan dalam waktu eksponensial terhadap ukuran KB.

Teorema ini dapat dibuktikan dengan menunjukkan bahwa himpunan klausa ALC (yaitu klausa yang diperoleh dari terjemahan ontologi ALC) yang digambarkan dalam Tabel 4.2 tertutup di bawah R_{ALC}. Dalam tabel, $\mathbf{P}(t)$, di mana t adalah suatu istilah, menunjukkan disjungsi yang mungkin kosong dalam bentuk $(\neg)P_1(t) \vee \cdots \vee (\neg)P_n(t)$. $\mathbf{P}(\mathbf{f}(x))$ menunjukkan disjungsi berbentuk $\mathbf{P}_1(f_1(x)) \vee \cdots \vee \mathbf{P}_m(f_m(x))$. Perhatikan bahwa definisi ini memungkinkan setiap $\mathbf{P}_i(f_i(x))$ untuk memuat literal positif dan negatif. Penulis mengadaptasi notasi klausa ALC dan membagi klausa tipe 2 karena literal yang dapat diselesaikan (dibahas di bawah) adalah tipe $P(f(x))$ jika ada dan jika tidak $P(x)$.

Ketika ontologi ALC dalam bentuk definitorial diterjemahkan ke klausa orde pertama seperti yang dijelaskan di bagian sebelumnya, setiap klausa adalah salah satu dari tipe yang tercantum dalam Tabel 4.2. Lebih lanjut, menerapkan resolusi terurut ke klausa dari tipe yang tercantum menghasilkan tipe klausa yang tercantum. Karena fungsi tidak bersarang, jumlah literal yang dapat dibangun menggunakan sejumlah simbol terbatas adalah terbatas. Oleh karena itu, jika literal yang duplikat dihapus, saturasi akan berakhir karena jika tidak, akan tercipta sejumlah klausa tak terbatas dari sejumlah literal yang terbatas. Pada bagian ini, penulis telah menjelaskan kalkulus resolusi yang valid dan lengkap untuk ALC. Yang perlu ditunjukkan selanjutnya adalah bahwa kita dapat mendefinisikan fungsi alokasi yang sesuai yang mempertahankan kelengkapan kalkulus ini.

Mendistribusikan resolusi ALC

Gagasan untuk mendefinisikan fungsi alokasi adalah untuk membatasi opsi inferensi sedemikian rupa sehingga terdapat literal unik untuk setiap klausa yang dapat diselesaikan dalam langkah inferensi selanjutnya dan menggunakan literal unik ini sebagai dasar untuk memutuskan alokasi klausa tersebut. Hal ini menjamin bahwa klausa yang dapat menjadi premis dari langkah resolusi selalu dikirim ke reasoner yang sama sehingga memungkinkan setiap inferensi yang terjadi di reasoner resolusi terpusat. Gagasan ini dinyatakan secara tepat dalam definisi dan hasil berikut.

Definisi 10. (Literal yang Dapat Diselesaikan) Sebuah literal L dari klausa $C \vee L$ adalah literal yang dapat diselesaikan dari $C \vee L$ sehubungan dengan R_{ALC} jika dan hanya jika terdapat klausa $D \vee L'$ sedemikian rupa sehingga R_{ALC} dapat diterapkan pada premis $C \vee L$ dan $D \vee L'$ yang menghasilkan klausa $(C \vee D)\sigma$ di mana σ adalah substitusi yang sesuai.

Prasyarat penting untuk strategi distribusi kita adalah bahwa untuk klausa tertentu, literal yang akan diselesaikan pada langkah resolusi berikutnya tidak bergantung pada kandidat premis lain yang tersedia, literal tersebut dapat ditentukan tanpa mempertimbangkan klausa lain.

Teorema 2. (Literal yang dapat diselesaikan secara unik) Setiap klausa ALC mengandung tepat satu literal yang dapat diselesaikan. Teorema 2 berlaku karena literal yang dapat diselesaikan harus dipilih atau maksimal dan merupakan bagian dari klausa tanpa literal yang dipilih. Klausa ALC mengandung paling banyak satu literal yang dipilih dan pengurutannya total pada klausa dasar dan klausa yang hanya mengandung satu variabel. Dengan demikian, hanya klausa yang mengandung lebih dari satu variabel yang dapat memiliki beberapa literal maksimal, tetapi kemudian klausa tersebut bertipe 3 dan mengandung satu literal yang dipilih.

Perhatikan bahwa untuk klausa ALC, menggunakan pengurutan LPO menghasilkan strategi sederhana untuk menentukan literal yang dapat diselesaikan: Jika klausa tersebut mengandung literal yang dipilih, maka literal inilah yang dapat diselesaikan. Jika tidak, jika klausa tersebut mengandung simbol fungsi, maka literal yang mengandung simbol fungsi tertinggi menurut prioritas simbol adalah literal yang dapat diselesaikan. Jika klausa tersebut tidak mengandung literal terpilih maupun fungsi, literal yang mengandung simbol predikat tertinggi adalah literal yang dapat diselesaikan.

Berdasarkan keunikan literal yang dapat diselesaikan, kita sekarang dapat mendefinisikan fungsi alokasi yang sesuai untuk ALC.

Definisi 11. (Alokasi untuk ASC) Alokasi klausa c untuk kalkulus terdistribusi RASC didefinisikan oleh

$$a_{ALC}(c) \coloneqq alloc\Big(topSymbol\big(resolvableLiteral(c)\big)\Big)$$

di mana

- $resolvableLiteral(c)$ adalah literal resolvable unik dari klausa c.
- $topSymbol(L) := P$ untuk setiap literal $L = (\neg)P(t)$ atau $L = (\neg)P(t_1, t_2)$ dengan simbol predikat unary atau biner P dan istilah t, t_1, t_2.
- $alloc : Sig(O) \to M$ adalah alokasi simbol tanda tangan dari ontologi masukan O, termasuk konsep yang diperkenalkan oleh transformasi bentuk definitorial.

Perhatikan bahwa untuk ontologi yang saling terkait, O adalah gabungan dari semua ontologi dan $alloc(X)$ dapat didefinisikan oleh namespace dari nama konsep atau properti X. Untuk ontologi tunggal, setiap partisi istilah ontologi menginduksi alokasi simbol melalui alokasi bagian secara acak ke modul. Dalam pekerjaan sebelumnya, penulis menunjukkan bahwa alokasi untuk ontologi monolitik dapat dihitung menggunakan metode partisi grafik.

Namun, jumlah komunikasi antar rekan lebih besar dan pemikir menghabiskan lebih banyak waktu menunggu input baru ketika basis pengetahuan adalah ontologi yang dipartisi karena bagian-bagian ontologi umumnya terhubung lebih padat daripada ontologi dalam jaringan ontologi. Untuk menentukan ke mana (dan apakah) klausa turunan dikirim, pertama-tama kita memilih literal yang dapat diselesaikan yang unik dari klausa tersebut, kemudian simbol teratas dari literal ini dan akhirnya pemikir yang dialokasikan simbol ini. Jika simbol s dialokasikan ke modul m, kita katakan bahwa modul m bertanggung jawab atas s. Gambar 4.2 mengilustrasikan resolusi terdistribusi pada contoh ontologi dari Gambar 4.1. Kelengkapan metode terdistribusi untuk memutuskan kepuasan ALC mengikuti dari sifat serupa R_{ALC}.

Teorema 3. (Kelengkapan resolusi ALC terdistribusi) Kalkulus resolusi terdistribusi $R_{ALC}(a_{ALC})$ adalah kalkulus resolusi lengkap untuk memutuskan kepuasan ASC.

Teorema 3 berlaku karena kita dapat menunjukkan bahwa batasan alokasi dipenuhi untuk setiap aplikasi aturan $r \in R_{ALC}$. Yaitu. Setiap pasangan klausa ALC yang dapat diselesaikan sebagai premis dalam inferensi resolusi terurut dialokasikan ke modul yang sama karena literal yang disatukan dalam inferensi memiliki predikat teratas yang sama dan merupakan literal yang dapat diselesaikan secara unik dari klausa masing-masing. Dengan demikian, mendistribusikan resolusi terurut sesuai dengan alokasi yang diberikan dalam Definisi 11 adalah metode terdistribusi yang sahih, lengkap, dan berakhir untuk memutuskan kepuasan ALC. Oleh karena itu, kita dapat mengalokasikan setiap klausa turunan ke reasoner berdasarkan namespace dari konsep atau nama properti tertentu. Teorema ini menjamin bahwa proses inferensi terdistribusi yang dihasilkan melakukan penalaran yang sahih dan lengkap atas ontologi yang saling terkait.

4.5 EVALUASI METODE RESOLUSI TERDISTRIBUSI

Penulis melakukan sejumlah eksperimen untuk menguji kinerja metode resolusi terdistribusi dalam pengaturan realistis yang dilaporkan di bagian ini. Tujuan utama dari eksperimen ini adalah untuk menganalisis efektivitas metode pada data realistis. Penulis mengidentifikasi tiga parameter yang memengaruhi efektivitas metode dan oleh karena itu merupakan target utama investigasi penulis selain waktu unggah dan waktu eksekusi sebenarnya.

Jumlah Klausa Turunan. Metode resolusi seringkali mengalami masalah dengan jumlah klausa yang sangat tinggi yang dihasilkan dalam proses derivasi. Hal ini terjadi karena resolusi bukanlah sesuatu yang didorong oleh tujuan, tetapi bergantung pada pembangkitan klausa baru dari basis pengetahuan yang ada hingga klausa kosong diturunkan, membangun sejumlah upaya pembuktian secara paralel. Dalam pengaturan terpusat, jumlah klausa dikurangi dengan sejumlah strategi untuk mengidentifikasi dan menghilangkan klausa yang berlebihan. Sudah diketahui bahwa penghapusan ini tidak dapat sepenuhnya dilakukan dalam pengaturan terdistribusi. Oleh karena itu, jumlah klausa yang dihasilkan oleh metode penulis dibandingkan dengan pengaturan terpusat merupakan parameter penting. Klaim penulis adalah bahwa jumlah klausa turunan tidak secara signifikan lebih tinggi daripada dalam pengaturan terpusat.

Jumlah Komunikasi. Beban komunikasi yang ditimbulkan oleh pertukaran klausa yang diperlukan antara reasoner merupakan sumber utama ketidakefektifan karena biaya komunikasi relatif tinggi dibandingkan dengan biaya untuk derivasi lokal. Hal ini sangat relevan jika reasoner didistribusikan dalam jaringan yang luas, yang merupakan pengaturan realistis dalam konteks web semantik. Secara umum, metode terdistribusi hanya dapat efektif jika jumlah komunikasi yang dibutuhkan kecil dibandingkan dengan langkah-langkah penalaran lokal. Oleh karena itu, penulis menyelidiki jumlah klausa yang dipertukarkan antara reasoner yang berbeda dibandingkan dengan jumlah keseluruhan klausa yang diturunkan. Penulis mengklaim bahwa jumlah klausa yang akan dipertukarkan bukanlah bagian yang signifikan dari

klausa yang diturunkan, atau dengan kata lain, sebagian besar klausa yang diturunkan tetap berada di reasoner yang sama tempat mereka diturunkan.

Tingkat Paralelisasi. Selain pengurangan jumlah data yang harus ditangani oleh satu reasoner, argumen utama yang mendukung penalaran terdistribusi adalah potensi untuk melakukan beberapa langkah penalaran secara paralel. Namun, ini membutuhkan skema komunikasi yang menjamin ketersediaan informasi yang diperlukan di semua reasoner. Dalam kasus skema komunikasi yang suboptimal, reasoner akan menganggur untuk sebagian atau sebagian besar waktu menunggu masukan dari reasoner lain. Hal ini akan secara signifikan mengurangi kinerja sistem secara keseluruhan. Oleh karena itu, penulis menyelidiki waktu menganggur rata-rata reasoner. Klaim penulis adalah bahwa semua reasoner sibuk hampir sepanjang waktu.

Implementasi

Implementasi resolusi terdistribusi penulis didasarkan pada pembukti orde pertama SPASSa, yang Algoritma 1 memberikan gambaran umumnya. Sejumlah strategi resolusi yang berbeda termasuk resolusi terurut didukung, prioritas dan seleksi dapat ditentukan dalam file input. Penulis mengimplementasikan normalisasi bentuk definitorial secara terpisah dan menggunakan Spass untuk klausifikasi ontologi yang dinormalisasi. Penulis menyimpan klausa dalam file terpisah dan menyertakan prioritas dan seleksi ke setiap file input.

Selain kepatuhan terhadap persyaratan RASC (Definisi 8), prioritasnya acak, fungsi seleksi memilih semua predikat biner negatif. Aturan reduksi yang diterapkan mencakup reduksi subsumsi maju dan mundur. Versi pra-rilis implementasi penulis tersedia secara online.c Untuk mengubah SPASS menjadi reasoner terdistribusi (yaitu menambahkan opsi "Terdistribusi"), ekstensi utamanya adalah dukungan untuk mengirim dan menerima klausa. Pengiriman adalah bagian yang mudah, metode pencetakan yang disediakan dimodifikasi untuk mencetak ke string alih-alih file dan string tersebut kemudian dikirim ke modul lain melalui koneksi TCP (*Transmission Control Protocol*).

Untuk menerima klausa baru, parser input dimodifikasi untuk membaca daftar klausa dari string yang kemudian ditambahkan ke penyimpanan klausa lokal. Sekumpulan klausa yang diterima diperlakukan seperti sekumpulan yang berasal dari klausa tertentu, yaitu direduksi maju dan mundur sehubungan dengan daftar klausa yang telah dikerjakan secara lokal sebelum menambahkan klausa yang diterima yang tidak redundan ke daftar yang dapat digunakan. Penting untuk memastikan bahwa reduksi maju dan mundur dilakukan sebelum mengirim dan setelah menerima klausa untuk menghapus sebanyak mungkin klausa redundan.

Karena jumlah modul kecil dalam pengaturan penulis, penulis menghubungkan semua reasoner saat startup, untuk jumlah modul yang lebih besar, penulis hanya akan membangun koneksi yang diperlukan. Untuk menghindari terblokirnya penalaran lokal saat mengirim klausa sementara modul tujuan sedang sibuk dan tidak dapat menerima, komunikasi klausa dilakukan dalam proses terpisah. Artinya, rutin pengiriman yang dipanggil setelah reduksi untuk klausa yang dialokasikan ke modul lain, sebenarnya tidak mengirim klausa tetapi menuliskannya ke buffer pengiriman. Proses komunikasi yang sebenarnya mengirimkan isi

buffer ini ke tujuannya ketika modul tujuan sedang menganggur. Prioritas proses penalaran dan komunikasi disesuaikan sedemikian rupa sehingga, meskipun tidak jenuh secara lokal, sebuah klausa hanya dikirim ke reasoner lain jika reasoner tujuan ini memberi sinyal bahwa ia sedang menganggur. Klausa baru hanya diterima ketika himpunan klausa lokal sepenuhnya jenuh.

Pengaktifan dan penonaktifan sistem diinisialisasi oleh proses kontrol pusat. Dalam sistem P2P yang sepenuhnya terdesentralisasi, tugas ini dilakukan oleh pihak yang menerima permintaan. Proses kontrol memulai mesin-mesin terpisah pada file klausa input masing-masing. Selain meneruskan klausa satu sama lain, reasoner mengirimkan pesan status setiap kali mereka jenuh secara lokal, ketika mereka melanjutkan penalaran pada klausa yang baru diterima, dan ketika mereka menghasilkan klausa kosong. Ketika satu reasoner menemukan bukti atau semua reasoner jenuh untuk interval yang lebih lama daripada waktu maksimal yang diperlukan untuk propagasi klausa, permintaan dijawab dan reasoner dimatikan.

Kumpulan Data

Penulis menguji implementasi penulis pada dua kumpulan data: Satu set dua ontologi besar yang dihubungkan oleh sejumlah besar tautan dan satu set lima ontologi yang lebih jarang terhubung. Kumpulan Data Anatomi. Ontologi dari jalur anatomi adalah Ontologi NCI yang menggambarkan anatomi manusia, yang diterbitkan oleh *National Cancer Institute* (NCI), dan Kamus Anatomi Tikus Dewasa, yang telah dikembangkan sebagai bagian dari proyek Basis Data Ekspresi Gen Tikus. Kedua sumber daya tersebut merupakan bagian dari *Open Biomedical Ontologies* (OBO). Tugas yang kompleks dan melelahkan dalam menghasilkan penyelarasan referensi telah dilakukan dengan kombinasi metode komputasi dan evaluasi manual yang ekstensif.

Tabel 4.3 Ontologi dari dataset OntoFarm yang digunakan dalam eksperimen.

Nama	***Jumlah Kelas***	***Jumlah Properti***	***Logika***
cmt	36	59	$ALCIF(D)$
conftool	38	36	$SIF(D)$
ekaw	77	33	$SHIN(D)$
iasted	140	41	$ALCIF(D)$
sigkdd	49	28	$ELI(D)$

Selain itu, ontologi diperluas dan diselaraskan untuk meningkatkan jumlah korespondensi antara kedua ontologi. Subset Ontologi NCI yang digunakan dalam eksperimen berisi 3304 konsep yang menggambarkan anatomi manusia, sedangkan ontologi anatomi tikus berisi 2744 konsep anatomi tikus. Meskipun satu ontologi menggambarkan manusia dan yang lainnya tikus, terdapat sejumlah besar korespondensi antara konsep dalam kedua ontologi tersebut. Pemetaan referensi yang dibuat secara manual berisi 1544 korespondensi antara konsep dalam kedua ontologi tersebut. Untuk memungkinkan menjawab pertanyaan, penulis menggunakan versi koheren dari penyelarasan referensi yang disediakan oleh penyelenggara *Ontology Alignment Evaluation Initiative* (OAEI). Kumpulan data ini tidak berisi individu,

melainkan mencakup skenario di mana sejumlah kecil ontologi yang relatif besar dengan daya ekspresi yang agak rendah dihubungkan melalui pemetaan.

Dataset OntoFarm. Dataset OntoFarm terdiri dari sekumpulan ontologi dalam domain organisasi konferensi yang telah dikumpulkan oleh para peneliti dari Kelompok Rekayasa Pengetahuan di Universitas Ekonomi Praha. Ontologi tersebut mencakup struktur konferensi, aktor yang terlibat, serta isu-isu yang terkait dengan proses pengajuan dan peninjauan. Dibandingkan dengan dataset anatomi, ontologi ini terhubung secara jarang dan relatif kecil. Dari 15 ontologi dalam koleksi ini, penulis menggunakan lima ontologi $(\mathrm{cmt}, \mathrm{confOf}, \mathrm{ekaw}, \mathrm{iasted}, \mathrm{sigkdd})$ dalam eksperimen penulis, karena untuk ontologi ini tersedia pemetaan yang dibuat dan diverifikasi secara manual. Tabel 4.3 merangkum properti dari kelima ontologi ini.

Agar dapat menerapkan metode penalaran terdistribusi penulis pada kumpulan ontologi ini, penulis mengkonversi properti tipe data menjadi properti objek dan menghilangkan operator logika di luar fragmen ALC, yaitu hierarki properti, properti fungsional, transitif, dan invers. Ontologi yang dihasilkan bersama dengan pemetaan yang dibuat dan diverifikasi secara manual digunakan sebagai dasar untuk serangkaian eksperimen kedua yang mencakup skenario sekumpulan ontologi kecil.

Eksperimen pada dataset anatomi

Penulis melakukan eksperimen pada Dataset Anatomi dalam tiga pengaturan berbeda. Pada pengaturan pertama, penulis menerjemahkan seluruh basis pengetahuan yang terdiri dari ontologi dan pemetaan yang dikodekan sebagai pernyataan kesetaraan antara predikat ke dalam klausa, memuat klausa-klausa ini ke dalam reasoner orde pertama SPASS, membiarkan reasoner memilih pengaturannya sendiri pada dataset. Penulis menyebut pengaturan ini sebagai `autoconfig' dan menggunakannya sebagai dasar. Penulis membandingkan pengaturan dasar ini dengan implementasi penulis sendiri dari metode resolusi terdistribusi yang diperkenalkan di atas. Lebih spesifiknya, penulis menggunakan dua pengaturan berbeda. Pengaturan pertama mensimulasikan pengaturan non-terdistribusi dengan hanya menggunakan satu peer reasoning yang menggunakan resolusi terurut untuk penalaran tentang seluruh basis pengetahuan. Penulis menyebut pengaturan ini sebagai `global'. Terakhir, penulis menggunakan pengaturan yang sesuai dengan visi penulis tentang penalaran terdistribusi.

Di sini, masing-masing dari dua ontologi ditempatkan pada peer penalaran. Aksioma kesetaraan yang mewakili pemetaan antara ontologi ditambahkan ke kedua ontologi. Resolusi terurut terdistribusi digunakan untuk menguji konsistensi basis pengetahuan secara keseluruhan. Penulis menyebut pengaturan ini sebagai 'terdistribusi'. Berikut ini penulis sajikan dan diskusikan hasil eksperimen penulis sehubungan dengan jumlah klausa turunan, jumlah komunikasi yang dibutuhkan, dan tingkat paralelisasi, serta waktu yang dibutuhkan untuk memuat dan melakukan penalaran tentang ontologi.

Waktu Pemuatan dan Eksekusi. Penulis membandingkan waktu yang dibutuhkan untuk memuat himpunan klausa ke dalam reasoner serta waktu yang dibutuhkan untuk menjalankan pemeriksaan konsistensi pada basis pengetahuan. Mengenai yang terakhir, perlu dicatat bahwa ontologi gabungan tersebut konsisten. Ini berarti bahwa basis pengetahuan harus

sepenuhnya jenuh dan tidak ada penghentian dini karena derivasi klausa kosong. Hasil perbandingan waktu unggah dan waktu eksekusi pada tiga pengaturan berbeda dirangkum dalam Tabel 4.4.

Tabel 4.4 Waktu unggah dan waktu eksekusi pada dataset Anatomi dalam detik.

	Autoconfig	**Global**	**Terdistribusi**
Update time	436.8	20.2	6.7
Runtime	515.8	202.4	55.1

Seperti yang diharapkan, memuat kumpulan klausa ke dalam reasoner membutuhkan waktu paling lama dalam pengaturan autoconfig. Ini wajar karena reasoner pertama-tama harus menganalisis data untuk menentukan konfigurasi parameter terbaik. Karena parameter ditentukan secara manual dalam pengaturan global, waktu unggah jauh lebih rendah. Kita juga dapat mengamati bahwa waktu unggah kembali berkurang dalam pengaturan terdistribusi. Hal ini dapat dijelaskan oleh fakta bahwa pada saat unggah, sejumlah struktur indeks diinisialisasi. Upaya untuk membuat indeks ini bersifat super-linier. Akibatnya, waktu yang dibutuhkan untuk mengunggah ontologi kembali berkurang secara signifikan dalam pengaturan terdistribusi meskipun jumlah keseluruhan klausa yang harus dimuat lebih tinggi karena klausa yang mewakili pemetaan antar ontologi diunggah dua kali. Dengan demikian, keuntungan pertama dari metode penalaran terdistribusi penulis adalah mengurangi waktu unggah untuk ontologi besar.

Kita dapat mengamati situasi serupa terkait dengan waktu eksekusi. Dalam pengaturan autoconfig, reasoner SPASS membutuhkan sembilan menit untuk memeriksa konsistensi basis pengetahuan secara keseluruhan. Dalam pengaturan global, ini sudah berkurang menjadi sekitar dua menit. Pengurangan ini dapat dijelaskan oleh pilihan metode resolusi spesifik yang diketahui berkinerja baik pada ontologi ALC.

Pengamatan pentingnya adalah bahwa dalam pengaturan terdistribusi, waktu eksekusi kembali berkurang secara signifikan menjadi kurang dari satu menit. Percepatan ini tidak hanya dapat dijelaskan oleh paralelisasi penalaran karena ini hanya dapat mengurangi waktu eksekusi menjadi sekitar 100 detik. Fakta bahwa kurang dari satu menit dibutuhkan menunjukkan bahwa pengaturan terdistribusi juga mendapat manfaat dari pengurangan ruang pencarian lokal yang membuat derivasi lokal lebih efektif. Ini berarti bahwa manfaat utama kedua dari metode penulis adalah percepatan penalaran lokal karena pengurangan ruang pencarian lokal.

Klausul Turunan dan Propagasi. Ketika mencoba menganalisis perilaku metode penalaran terdistribusi penulis dan menjelaskan pengurangan waktu eksekusi yang signifikan, pertama-tama kita melihat jumlah klausa turunan dan jumlah klausa yang dipropagasi antara rekan-rekan penalaran yang berbeda. Tabel 4.5 merangkum hasilnya sehubungan dengan aspek ini.

Tabel 4.5 Derivasi dan propagasi klausa pada dataset Anatomi dalam jumlah klausa turunan.

	Autoconfig	**Global**	**Terdistribusi**
Derivasi	15324	15324	15324
Propagasi	–	–	–
	(0)%	(0)%	(56)%

Tabel 4.6 Tingkat paralelisasi pada dataset Anatomi dalam waktu sibuk rata-rata dari rekan penalaran.

	Autoconfig	**Global**	**Terdistribusi**
Runtime	536.2	112.4	55.1
Rata-rata Waktu sibuk	536.2	112.4	45.0
	(100)%	(100)%	(82)%

Hal pertama yang kita perhatikan adalah fakta bahwa dalam ketiga pengaturan tersebut, jumlah klausa yang sama diturunkan untuk menghitung penutupan basis pengetahuan. Untuk pengaturan terdistribusi, ini adalah kabar baik karena berarti bahwa untuk pengaturan ini cukup untuk melakukan pemeriksaan redundansi dan reduksi pada rekan lokal tanpa kehilangan potensi optimasi yang berpotensi menjadi masalah utama untuk metode pembuktian teorema terdistribusi.

Kita dapat melihat lebih lanjut bahwa sejumlah besar komunikasi terjadi antara dua rekan penalaran dalam pengaturan terdistribusi. Ini berarti bahwa hasil yang baik terkait waktu eksekusi tidak dapat dijelaskan hanya oleh kurangnya interaksi antara ontologi. Lebih lanjut, ini menunjukkan bahwa upaya yang dibutuhkan untuk bertukar klausa antara penalaran diimbangi oleh manfaat memiliki ruang pencarian lokal yang lebih kecil.

Tingkat Paralelisasi. Selanjutnya, kita melihat tingkat paralelisme dengan melihat waktu rata-rata rekan penalaran sibuk dibandingkan dengan waktu eksekusi keseluruhan sistem. Jelas, dalam konfigurasi otomatis dan pengaturan global di mana kita hanya memiliki satu penalaran, penalaran ini aktif sepanjang waktu dan karenanya sibuk selama 100%. Parameter yang menarik adalah waktu rata-rata yang dihabiskan oleh peer yang melakukan penalaran dalam pengaturan terdistribusi. Informasi terkait disajikan dalam Tabel 4.6.

Seperti yang dapat kita lihat dari Tabel, tingkat paralelisme yang cukup baik telah dicapai pada dataset anatomi. Selama waktu eksekusi keseluruhan 55 detik, masing-masing dari dua reasoner sibuk selama rata-rata 45 detik. Ini adalah tingkat lebih dari 80% yang menunjukkan bahwa memang sebagian besar komputasi dieksekusi secara paralel yang menyebabkan pengurangan lebih lanjut dari waktu eksekusi keseluruhan.

Eksperimen pada dataset OntoFarm

Serangkaian eksperimen kedua dilakukan pada lima ontologi dari Dataset OntoFarm yang disebutkan di atas. Tujuan dari eksperimen kedua ini adalah untuk menganalisis perilaku metode pada sekumpulan ontologi kecil yang saling terhubung melalui pemetaan, yang merupakan situasi khas lain yang dihadapi di web semantik. Sebagai pengaturan untuk eksperimen ini, penulis mempertimbangkan ontologi dalam urutan abjad dimulai dengan ontologi CMT, menambahkan satu ontologi demi satu termasuk pemetaan ke semua ontologi

yang telah dipertimbangkan sebelumnya. Statistik dari berbagai konfigurasi yang dihasilkan diberikan dalam Tabel 4.7.

Penulis menjalankan metode penalaran penulis pada ontologi yang dihasilkan dan membandingkan konfigurasi lokal dengan konfigurasi terdistribusi. Dalam percobaan ini, penulis tidak lagi mempertimbangkan pengaturan autoconfig karena percobaan pertama jelas menunjukkan bahwa mengkonfigurasi reasoner ke resolusi terurut berkinerja jauh lebih baik daripada pengaturan generik. Lebih lanjut, penulis tidak mempertimbangkan waktu unggah sebagai kriteria karena ontologi yang terlibat relatif kecil, berkisar antara dua hingga enam ratus klausa yang tidak cukup untuk menghasilkan angka yang bermakna. Selain perbedaan ini, penulis kembali mengukur parameter yang diidentifikasi dalam pengantar bagian ini, yaitu waktu eksekusi, jumlah klausa turunan, jumlah klausa yang disebarkan, dan waktu sibuk rata-rata dari peer penalaran.

Selain percobaan pertama, penulis juga melihat bagaimana parameter ini berubah dengan meningkatnya jumlah ontologi dan peer penalaran yang merupakan masalah penting terkait dengan skalabilitas. Lebih lanjut, penulis mencatat waktu eksekusi dan waktu sibuk tidak hanya untuk uji konsistensi tetapi juga untuk kueri subsumsi positif. Kueri negatif (yaitu kueri yang dijawab dengan "Tidak") tidak banyak bervariasi dalam waktu eksekusi, hasilnya dapat diharapkan mirip dengan uji konsistensi. Kueri subsumsi menguji apakah aksioma $A_1 \sqsubseteq A_2$ yang diberikan dapat diturunkan dari jaringan ontologi dengan menambahkan klausa asertif $A_1(testinstance)$ dan $\neg A2(testinstance)$ ke himpunan klausa yang sesuai.

Jika aksioma tersebut mengikuti dari jaringan ontologi, kueri tersebut positif dan klausa kueri menghasilkan kontradiksi. Untuk membuat semua kueri positif, perlu untuk mengklasifikasikan (yaitu menghitung hierarki konsep) ontologi terlebih dahulu. Jika tidak, semua pasangan konsep harus diuji yang akan memakan waktu sekitar dua hari. Karena penulis belum mengimplementasikan klasifikasi untuk metode resolusi terdistribusi penulis, penulis menggunakan penalaran tabel umum Pellete untuk klasifikasi.

Mengklasifikasikan seluruh jaringan ontologi yang digabungkan dengan pemetaan gagal karena keterbatasan memori, oleh karena itu penulis hanya menghasilkan semua kueri subsumsi positif dari dua jaringan ontologi yang terdiri dari cmt, confof, ekaw dan iasted, sigkdd masing-masing. Untuk setiap dari lima jaringan ontologi yang penulis gunakan untuk uji konsistensi, penulis juga menguji kueri subsumsi positif yang berlaku yang diperoleh melalui klasifikasi dan menghitung waktu eksekusi rata-rata dan waktu sibuk.

Tabel 4.7 Ukuran dataset yang digunakan dalam eksperimen dalam hal jumlah klausa.

	Jumlah Klausa yang dipetakan ke					
	Ontologi	cmt	confof	ekaw	iasted	Total
cmt	228					228
confof	199	24				451
ekaw	256	19	40			766
Iasted	587	8	18	20		1399
sigkdd	153	20	13	22	30	1637

Waktu Eksekusi. Aspek pertama dan terpenting adalah waktu eksekusi keseluruhan sistem sehubungan dengan pengecekan konsistensi basis pengetahuan secara keseluruhan. Seperti pada kasus anatomi, ontologi yang dipertimbangkan konsisten, yang berarti bahwa basis pengetahuan harus sepenuhnya jenuh untuk uji konsistensi. Hasil perbandingan waktu eksekusi keseluruhan sistem dalam pengaturan global di mana semua inferensi dilakukan oleh satu peer penalaran dan pengaturan terdistribusi di mana setiap ontologi ditugaskan ke peer penalaran khusus dibandingkan dalam Tabel 4.8.

Tabel 4.8 Waktu eksekusi pada dataset OntoFarm dalam detik.

	Saturasi		**Kueri Positif**	
	Global	Terdistribusi	Global	Terdistribusi
cmt	0.24	0.24		
cmt + confof	0.32	0.24	0.04	0.05
cmt + confof + ekaw	1.09	0.52	0.06	0.09
cmt + confof + ekaw + iasted	11.22	2.54	0.20	0.12
cmt + confof + ekaw + iasted + sigkdd	13.50	3.59	0.24	0.15

Seperti yang dapat kita lihat, hasil eksperimen mengkonfirmasi pengamatan eksperimen anatomi dalam arti bahwa kita mengamati pengurangan signifikan dalam waktu eksekusi keseluruhan ketika mendistribusikan proses inferensi uji konsistensi. Meskipun pengurangan ini relatif kecil untuk dataset kecil, pengurangan ini menjadi lebih signifikan semakin banyak ontologi dan rekan penalaran yang terlibat: sementara untuk tiga ontologi pengaturan terdistribusi sekitar dua kali lebih cepat daripada pengaturan global, dalam kasus empat ontologi terdapat peningkatan kecepatan hampir lima kali lipat dibandingkan dengan pengaturan global.

Peningkatan kecepatan ini dapat dijelaskan lagi oleh pengurangan ruang pencarian lokal yang mengurangi waktu untuk derivasi lokal dan mengkompensasi upaya komunikasi tambahan yang dalam hal ini jauh lebih tinggi daripada kasus anatomi seperti yang akan kita lihat selanjutnya. Untuk kueri positif, seperti yang diharapkan, waktu eksekusi absolut dan juga perbedaan antara pengaturan global dan terdistribusi jauh lebih kecil. Di sini, komputasi terdistribusi hanya lebih cepat jika lebih dari tiga rekan penalaran terlibat. Untuk membandingkan metode penulis dengan penalaran tableau umum, penulis menjalankan Pellet pada ontologi gabungan yang terdiri dari semua modul dan pemetaan ontologi. Pemeriksaan konsistensi membutuhkan waktu sekitar 3 detik, jawaban untuk kueri positif acak dikembalikan setelah 1,7 detik.

Jumlah Klausa Turunan. Melihat jumlah klausa turunan, kita melihat efek dari ontologi yang lebih ekspresif. Sementara dalam kasus anatomi, evaluasi terdistribusi tidak berdampak pada jumlah keseluruhan klausa turunan, hal ini tidak lagi berlaku pada dataset OntoFarm. Hasil eksperimen terkait masalah ini dirangkum dalam Tabel 4.9.

Tabel 4.9 Jumlah klausa turunan pada dataset OntoFarm

Ontologi	**Global**	**Terdistribusi**

cmt	2031	2031
cmt + confof	3726	3752
cmt + confof + ekaw	9586	9680
cmt + confof + ekaw + iasted	35809	53935
cmt + confof + ekaw + iasted + sigkdd	41023	74028

Kita melihat bahwa untuk sejumlah kecil ontologi (hingga tiga), jumlah keseluruhan klausa turunan hanya meningkat sedikit, tetapi untuk dua pengaturan terakhir dengan empat dan lima ontologi, sejumlah besar derivasi tambahan dapat diamati. Kemungkinan besar perilaku ini akan selalu diamati dalam sistem dengan banyak ontologi. Saat ini, mekanisme reduksi diterapkan pada klausa baru sebelum dikirim dan setelah diterima oleh rekan penalaran. Selama interaksi terutama terjadi antara pasangan ontologi, strategi ini tampaknya efisien. Ketika lebih banyak ontologi terlibat, interaksi akan terjadi antara beberapa ontologi, yang tidak semuanya dapat dideteksi selama komunikasi bilateral. Meskipun peningkatan derivasi yang dihasilkan diimbangi oleh penghematan karena ruang pencarian lokal yang lebih kecil dalam eksperimen penulis, strategi yang lebih efektif untuk mendeteksi klausa redundan sehingga mengurangi jumlah klausa turunan merupakan masalah utama untuk pekerjaan di masa mendatang.

Jumlah Komunikasi. Mengenai jumlah komunikasi antar rekan penalaran, situasinya mirip dengan pengamatan yang dilakukan sehubungan dengan jumlah keseluruhan klausa turunan. Berbeda dengan yang terakhir, penulis telah mengamati kebutuhan akan komunikasi yang signifikan antar rekan penalaran dalam eksperimen anatomi. Dibandingkan dengan jumlah komunikasi dalam eksperimen tersebut, fraksi klausa yang harus dikirim lebih kecil. Angka konkretnya diberikan dalam Tabel 4.10.

Seperti yang ditunjukkan Tabel 4.10, jumlah komunikasi yang dibutuhkan untuk memenuhi basis pengetahuan meningkat seiring dengan jumlah ontologi yang terlibat. Kita melihat peningkatan yang signifikan dari tiga hingga empat ontologi yang dapat dijelaskan oleh peningkatan tajam jumlah klausa turunan secara keseluruhan. Selain efek ini, fraksi klausa yang harus dikirim juga meningkat seiring dengan jumlah ontologi. Dibandingkan dengan kasus anatomi di mana lebih dari setengah klausa dikirim dalam pengaturan terdistribusi, fraksi ini lebih kecil untuk dataset OntoFarm. Hal ini dapat dijelaskan oleh fakta bahwa terdapat lebih banyak tautan antar entitas dalam ontologi anatomi dan derivasi pengetahuan baru hampir selalu terdiri dari penggabungan pengetahuan yang ada dari kedua ontologi tersebut. Karena ontologi dalam dataset OntoFarm lebih jarang saling terkait, sebagian besar pengetahuan baru sudah dapat diturunkan di dalam ontologi individual sehingga menurunkan jumlah relatif pengetahuan yang dipertukarkan antar ontologi.

Penggunaan strategi reduksi yang lebih efektif seperti yang dijelaskan di atas juga akan mengurangi jumlah klausa yang akan dikirim. Di luar itu, kecil kemungkinan untuk meningkatkan angka yang diberikan karena kelengkapan metode penalaran penulis bergantung pada penugasan langkah inferensi yang tidak ambigu kepada rekan penalaran dan eksperimen sebelumnya menunjukkan bahwa penugasan inferensi kepada penalaran

berdasarkan teknik partisi tidak seefisien hasil yang dicapai dengan menggunakan distribusi pengetahuan alami yang merupakan dasar dari hasil dalam pada bab ini ini.

Tabel 4.10 Jumlah klausa yang disebarkan pada dataset OntoFarm

Ontologi	Global	Terdistribusi	
cmt	0	0	(0)%
cmt + confof	0	288	(8)%
cmt + confof + ekaw	0	1466	(15)%
cmt + confof + ekaw + iasted	0	14624	(27)%
cmt + confof + ekaw + iasted + sigkdd	0	25217	(34)%

Tingkat Paralelisasi. Ketika lebih banyak ontologi terlibat, masalah paralelisasi menjadi kritis karena langkah-langkah inferensi dalam satu peer penalaran dapat bergantung pada derivasi yang terjadi di reasoner lain tanpa jaminan bahwa hasil yang dibutuhkan tersedia tepat waktu bagi reasoner lain untuk melanjutkan. Hasil investigasi masalah ini ditunjukkan pada Tabel 4.11.

Hasil eksperimen penulis menunjukkan bahwa tingkat paralelisasi yang dicapai pada uji konsistensi dataset OntoFarm tidak mencapai hasil pada dataset anatomi di mana waktu sibuk lebih dari 80% diamati. Dalam eksperimen ini, penulis mengamati waktu sibuk rata-rata sekitar 60% untuk semua pengaturan. Ini lebih rendah daripada untuk dataset anatomi tetapi masih merupakan hasil yang dapat diterima, terutama karena waktu sibuk rata-rata tampaknya tidak menurun dengan jumlah ontologi. Ini berarti bahwa sistem secara keseluruhan menjadi lebih efektif dengan bertambahnya jumlah peer karena peningkatan jumlah peer berkontribusi pada solusi masalah yang bersama dengan waktu sibuk rata-rata yang stabil menyebabkan peningkatan linier daya komputasi yang tersedia untuk memecahkan masalah. Lebih lanjut, tingkat paralelisasi mendekati 100% untuk kueri positif. Ini tidak mengejutkan, karena bukti untuk kueri ditemukan jauh sebelum uji konsistensi dijawab, pada saat sebagian besar rekan penalaran belum menyelesaikan saturasi himpunan klausa awal mereka. Hasil ini memang penting karena mendukung klaim penulis bahwa metode yang disajikan dapat diskalakan untuk masalah yang lebih besar.

Tabel 4.11 Rata-rata waktu sibuk rekan penalaran dalam pengaturan terdistribusi.

Ontologi	Saturasi		Kueri Positif	
cmt	0.24	(100)%		
cmt + confof	0.14	(59)%	0.05	(99)%
cmt + confof + ekaw	0.28	(54)%	0.09	(99)%
cmt + confof + ekaw + iasted	1.70	(67)%	0.12	(99)%
cmt + confof + ekaw + iasted + sigkdd	2.00	(56)%	0.14	(92)%

Ringkasan Hasil

Singkatnya, eksperimen penulis dengan jelas menunjukkan bahwa metode penalaran terdistribusi yang diusulkan dalam pada bab ini ini secara konsisten mengungguli komputasi

global baik dalam hal waktu unggah maupun waktu eksekusi. Secara khusus, penulis mengukur peningkatan kecepatan eksekusi hingga empat kali lipat untuk lima rekan penalaran. Pengamatan lebih dekat pada proses inferensi mengungkapkan bahwa komunikasi yang diperlukan dalam pengaturan terdistribusi sangat signifikan — terkadang lebih dari setengah klausa yang diturunkan dikirim ke penalaran lain — tetapi biaya komunikasi tersebut diimbangi oleh pengurangan ruang pencarian lokal yang mengarah pada inferensi lokal yang lebih cepat dan peningkatan linier daya komputasi yang tersedia untuk memecahkan masalah dengan eksekusi paralel dari proses inferensi.

4.6 ARAH PENGEMBANGAN METODE RESOLUSI TERDISTRIBUSI

Berdasarkan hasil eksperimen, penulis telah mengidentifikasi pengurangan efektif klausa redundan sebagai target utama untuk optimasi lebih lanjut karena dengan adanya beberapa rekan penalaran, jumlah derivasi secara keseluruhan meningkat karena penghapusan klausa redundan yang tidak lengkap. Selain itu, implementasi yang digunakan untuk evaluasi eksperimental akan diperluas dalam beberapa cara. Pertama, ekspresivitas yang didukung dapat diperluas ke $ALCHI$ tanpa kehilangan kelengkapan prosedur penalaran. Selanjutnya, kinerja pendekatan penulis dapat ditingkatkan dengan mengoptimalkan alokasi simbol dan klausa secara dinamis. Terakhir, menyelidiki variasi dalam prioritas simbol yang menentukan urutan literal dapat mengurangi waktu eksekusi dan upaya komunikasi.

Perluasan ke logika deskripsi yang lebih ekspresif

Investigasi teoretis dan evaluasi praktis yang disajikan dalam karya ini menerapkan resolusi terdistribusi ke ontologi ALC. Pada bagian ini, kita membahas penerapan metode penulis ke ontologi yang lebih ekspresif.

Transitivitas. Aksioma transitivitas (misalnya, Trans(partOf)) digunakan untuk menyatakan bahwa properti tertentu bersifat transitif. Aksioma ini dapat dihilangkan dengan transformasi yang terkenal, mengurangi ekspresivitas suatu ontologi menjadi ALC. Oleh karena itu, dengan beberapa pra-pemrosesan, kita dapat menentukan kepuasan ontologi ALC dengan aksioma transitivitas, yaitu logika deskripsi S. Transformasi ini bersifat polinomial dalam ukuran input, tetapi adaptasi ke pengaturan terdistribusi bukanlah hal yang sepele. Berbeda dengan transformasi aksioma logika deskripsi ke klausa orde pertama yang telah disebutkan sebelumnya, translasi transitivitas bergantung pada keseluruhan ontologi dan bukan hanya pada aksioma transitivitas. Oleh karena itu, ontologi yang terhubung tidak dapat ditransformasikan secara independen.

Nominal. Nominal adalah konsep dengan satu contoh, misalnya, konsep $\{Erdo€s\}$ dengan contoh $Erdo€s$ digunakan dalam deskripsi konsep $E\ coauthorOf\ \{Erdo€s\}$. Nominal digantikan oleh konsep umum untuk banyak aplikasi: Setiap nominal $\{nom\}$ digantikan oleh konsep baru Nom dan aksioma $Nom(nom)$ ditambahkan ke ontologi. Batasan bahwa Nom tidak boleh berisi contoh lain tidak dapat diungkapkan dalam logika deskripsi tanpa nominal.

Tipe data. Tipe data (D) dapat dihilangkan tanpa mengubah semantik dengan memindahkan tipe data ke domain abstrak. Dalam praktiknya, jenis (tipe data dan abstrak)

ditangani berbeda dari predikat lain untuk mempercepat penalaran. Predikat tipe data bawaan dapat ditambahkan untuk mendukung misalnya, relasi yang lebih besar antara bilangan bulat.

Pembatasan Kardinalitas. Pembatasan kardinalitas digunakan untuk menyatakan properti tertentu sebagai fungsional atau mendefinisikan misalnya, orang kaya adalah orang yang memiliki setidaknya tiga rumah ($RichPerson \sqsubseteq Person \sqcap \exists_{\geq 3} owns.House$). Jika ontologi berisi pembatasan kardinalitas yang memenuhi syarat (Q) atau tidak memenuhi syarat (N) atau properti fungsional (F), himpunan klausa orde pertama yang sesuai berisi literal kesamaan. Untuk menangani kesamaan ini, kalkulus yang lebih rumit daripada resolusi terurut diperlukan untuk memutuskan kepuasan. Kalkulus yang memutuskan kepuasan untuk klausa yang diperoleh dari ontologi $ALCHIQ$ adalah kalkulus superposisi. Namun, aturan superposisi dapat diterapkan pada premis di mana literal yang dapat diselesaikan memiliki simbol atas yang berbeda. Oleh karena itu, perluasan pendekatan penulis ke arah ini tidak mudah. Resolusi terdistribusi untuk topik kardinalitas.

Alokasi Fleksibel

Dalam pekerjaan ini, penulis mengasumsikan alokasi simbol ditentukan oleh namespace dari istilah ontologi. Dalam pekerjaan sebelumnya, penulis menunjukkan bahwa alokasi untuk ontologi monolitik dapat dihitung menggunakan metode partisi grafik. Dalam kedua pengaturan tersebut, alokasi ditentukan sebelum proses penalaran, tetapi dimungkinkan untuk memperluas metode tersebut ke alokasi simbol yang lebih fleksibel. Alokasi fleksibel memiliki dua keuntungan untuk kinerja penalaran: Pertama, beban komputasi dapat didistribusikan di antara rekan-rekan secara lebih merata. Jika satu rekan sangat sibuk sementara yang lain tetap menganggur, simbol yang dialokasikan ke rekan yang sibuk dapat dialokasikan kembali ke rekan yang menganggur. Kedua, komunikasi dapat dikurangi dengan mengoptimalkan alokasi secara dinamis dengan tepat. Secara teori, alokasi simbol dapat diubah pada titik mana pun dalam proses penalaran. Prosedur alokasi ulang mirip dengan pembaruan dalam tabel perutean. Meskipun pembaruan bersifat asinkron dan rekan-rekan mungkin untuk sementara menggunakan alokasi yang berbeda, klausa tidak pernah dihapus secara tidak sengaja karena rekan-rekan yang menerima klausa yang bukan tanggung jawab mereka hanya akan meneruskannya sesuai dengan alokasi lokal mereka. Untuk menghindari penyebaran klausa bolak-balik, peer Baru yang bertanggung jawab atas simbol P yang dialokasikan ulang harus menjadi yang pertama memperbarui tabel alokasi lokal. Kemudian peer Lama yang bertanggung jawab atas P sebelumnya mengubah tabelnya dan mengirimkan semua klausa dengan simbol literal teratas P yang dapat diselesaikan ke Baru. Peer lain diperbarui selanjutnya, jika Lama menerima klausa yang seharusnya dialokasikan ke Baru, klausa tersebut diteruskan. Dimungkinkan juga untuk memperkenalkan peer penalaran baru dengan mengalokasikan ulang simbol atau mematikan peer dan mengalokasikan ulang set klausa lokalnya ke peer lain.

Prioritas Lanjutan

Prioritas simbol yang menentukan urutan literal dan dengan demikian juga memengaruhi alokasi klausa harus didefinisikan sedemikian rupa sehingga simbol fungsi mendahului simbol predikat dan negasi adalah simbol terkecil. Terlepas dari persyaratan ini,

prioritas dalam eksperimen penulis bersifat acak. Karena daftar simbol digabungkan dari file terpisah, prioritasnya adalah penggabungan daftar fungsi dan predikat terpisah dari semua ontologi. Konfigurasi ini dapat dilihat sebagai dasar, mengoptimalkan prioritas modul dan prioritas lokal dapat mengurangi jumlah propagasi dan waktu eksekusi keseluruhan.

4.7 KESIMPULAN DAN ARAH PENGEMBANGAN PENALARAN TERDISTRIBUSI

Dalam buku ini, penulis telah membahas masalah pendistribusian metode penalaran untuk ontologi di berbagai reasoner di web. Fokus penulis adalah pada pendistribusian penalaran resolusi dan penulis menunjukkan bahwa resolusi dapat didistribusikan tanpa kehilangan kelengkapan jika klausa yang dapat diselesaikan selalu diproses oleh reasoner yang sama. Berdasarkan pengamatan ini, penulis menyajikan metode resolusi terdistribusi yang valid dan lengkap untuk ontologi yang dibangun berdasarkan karya yang ada tentang penalaran resolusi untuk ontologi ALC. Pada bab ini dapat dilihat sebagai langkah pertama menuju arsitektur penalaran terdistribusi yang sangat skalabel untuk web semantik dan sejalan dengan upaya terbaru dalam membangun infrastruktur penalaran tersebut. Pekerjaan selanjutnya akan berkaitan dengan perluasan pendekatan ke bahasa ontologi yang lebih ekspresif serta dengan optimasi pendekatan penalaran dengan menyelidiki urutan tanda tangan ontologi yang meminimalkan overhead komunikasi.

BAB 5
DETEKSI ADEGAN VIDEO MULTIMODAL

Deteksi adegan video adalah tugas membagi video menjadi bagian-bagian semantik. Untuk melakukan tugas mendasar ini, penulis mengusulkan metode baru dan efektif untuk pengelompokan temporal adegan menggunakan serangkaian fitur arbitrer yang dihitung dari video. Penulis merumuskan tugas deteksi adegan video sebagai masalah optimasi generik untuk mengelompokkan bidikan secara optimal ke dalam adegan, dan mengusulkan prosedur efisien untuk menyelesaikan masalah optimasi berdasarkan skema pemrograman dinamis baru. Formulasi unik ini secara langsung menghasilkan segmentasi yang konsisten secara temporal, dan memiliki keunggulan tanpa parameter, sehingga dapat diterapkan di berbagai domain. Penulis memberikan hasil eksperimen terperinci, yang menunjukkan bahwa algoritma penulis mengungguli metode-metode terkini. Untuk menilai kelengkapan metode ini lebih lanjut, penulis menyajikan hasil eksperimen yang menguji berbagai jenis modalitas dan penerapannya dalam formulasi ini.

5.1 DETEKSI ADEGAN VIDEO BERBASIS STRUKTUR SHOT DAN MULTIMODAL

Saat berurusan dengan konten video, penting untuk memahami struktur hierarkis yang melekat yang menyusunnya. Bingkai individual yang membentuk keseluruhan video adalah tingkat terendah dari hierarki tersebut. Urutan bingkai yang diambil dari kamera yang sama pada periode waktu tertentu disebut bidikan (shot). Tingkat pembagian semantik di atas bidikan membagi video menjadi adegan, yang merupakan tujuan dari bab ini. Penulis mendefinisikan adegan sebagai urutan bidikan yang terkait secara semantik dan berdekatan secara temporal yang menggambarkan konsep atau cerita tingkat tinggi. Deteksi adegan video, tugas membagi video menjadi adegan-adegan penyusunnya, sangat penting ketika mencoba untuk menyimpulkan semantik dan pengaturan adegan video, atau melakukan jenis pemahaman semantik lainnya dalam video heterogen. Untuk tujuan ini, tugas deteksi adegan video merupakan tahap pra-pemrosesan penting untuk membagi video yang lebih panjang dengan banyak adegan menjadi segmen yang koheren secara semantik. Selain itu, untuk video yang sangat panjang, deteksi adegan video dapat menjadi penting untuk peringkasan video dan sebagai alat pengindeksan untuk memungkinkan penelusuran dan pengambilan bagian video yang relevan dengan cepat.

Pendekatan umum untuk deteksi adegan video adalah dengan mengandalkan pembagian menjadi beberapa bidikan, dan kemudian mengelompokkan bidikan tersebut ke dalam adegan dengan menggunakan berbagai metode pengelompokan. Hal ini yang dilakukan berkali-kali dapat menyebabkan masalah yang penulis sebut inkonsistensi temporal di mana algoritma pembagian akan mencoba untuk menetapkan bidikan yang tidak berdekatan secara temporal ke dalam satu adegan. Bahkan jika kedekatan temporal diperhitungkan, tidak ada jaminan bahwa dua bidikan yang sangat mirip di sisi berlawanan dari video tidak akan digabungkan ke dalam satu adegan meskipun jarak temporalnya.

Untuk mengatasi hal ini, penulis mengusulkan metode yang membagi video sambil memanfaatkan struktur inheren dari urutan bidikan. Beberapa metode yang ada saat ini juga dapat membagi video dengan cara ini. Namun, dengan metodologi penulis, pendekatan yang lebih umum diimplementasikan yang dapat diterapkan pada berbagai jenis konten video dan terbukti mengungguli metode terkini. Penggunaan pemrograman dinamis membuat metode penulis efisien dan memiliki waktu eksekusi yang praktis. Selain itu, fakta bahwa metode penulis tidak bergantung pada parameter menghilangkan kebutuhan untuk melakukan penyesuaian pada dataset tertentu, menjadikannya metode ideal untuk deteksi adegan video secara langsung.

Saat menentukan cara memanfaatkan informasi dari sebuah video, akan bermanfaat untuk mendapatkan fitur yang berbeda dari lebih dari satu modalitas. Setiap jenis modalitas pasti akan berkinerja berbeda, dan untuk berbagai jenis konten, modalitas tertentu akan lebih berguna daripada yang lain. Misalnya, dalam film, konten visual atau akustik mungkin akan menyimpan informasi yang paling membedakan, sementara untuk wawancara studio mungkin perlu menggunakan transkripsi ucapan. Untuk menyesuaikan setiap fitur ke dalam formulasi, sangat penting untuk merepresentasikan data dan memilih metrik jarak sedemikian rupa sehingga fitur intra-adegan menghasilkan jarak kecil sementara fitur dari adegan yang berbeda menghasilkan jarak yang besar.

5.2 METODE SEGMENTASI DAN DETEKSI ADEGAN VIDEO

Bagian ini memberikan tinjauan beberapa metode terbaru. Kami membatasi diri untuk mempertimbangkan metode yang mencoba melakukan segmentasi temporal lengkap dari sebuah video (sebagai lawan dari ekstraksi adegan) dan yang didasarkan pada fitur visual.

Deteksi batas adegan

Membagi sebuah video menjadi adegan-adegan penyusunnya dianggap sebagai langkah awal yang penting untuk deteksi adegan video. Namun, karena definisi adegan yang sangat tepat dan objektif serta keseragaman bingkai dalam sebuah adegan, hal ini biasanya dianggap sebagai masalah yang sudah terpecahkan. Berbagai metode efektif untuk deteksi batas adegan ada. Penulis menggunakan kesamaan histogram dan pencocokan fitur untuk mengukur jarak antar bingkai, dan kemudian mereka menunjukkan batas di mana jarak antara bingkai yang berdekatan cukup besar.

Mereka juga mengukur jarak per-frame sebagai fungsi waktu untuk mengidentifikasi rentang yang lebih panjang di mana terdapat jarak yang besar antara frame, untuk menemukan transisi bertahap antara bidikan. Jarak piksel dan histogram digunakan. Mereka mendeteksi transisi bertahap dengan mengukur jarak antara frame dengan celah yang meningkat melampaui frame yang berdekatan. Selama eksperimen penulis, penulis menggunakan versi yang sedikit dimodifikasi dengan jarak yang berbeda diberi bobot secara probabilistik menggunakan pencocokan histogram.

Penulis mencatat bahwa meskipun video bidikan tunggal tidak dapat dibagi menjadi bidikan, video tersebut mungkin terdiri dari sejumlah adegan. Contoh yang baik adalah video rumahan dari kamera genggam, yang bisa sangat panjang dan tanpa jeda. Untuk kasus ini,

karena transisi adegan yang tepat sangat ambigu, cukup untuk menandai batas bidikan setiap beberapa detik dan melanjutkan deteksi adegan sesuai kebutuhan.

Deteksi adegan video

Pada bagian ini, penulis meninjau sejumlah metode untuk deteksi adegan video. Penulis mengukur kesamaan antara bidikan dengan menghitung kesamaan warna dan gerakan yang dibantu oleh key-frame. Mereka membuat grafik kemiripan adegan, di mana bobotnya adalah kemiripan yang dihitung yang dibobot secara terbalik oleh jarak temporal antar adegan. Pemotongan yang dinormalisasi dilakukan secara rekursif hingga kriteria penghentian terpenuhi, hanya mengizinkan pemotongan yang membagi sepanjang sumbu temporal.

Adegan dikelompokkan hanya berdasarkan kemiripan visual. Penyelarasan urutan dilakukan untuk mengisolasi bagian-bagian adegan yang berdekatan secara temporal yang menampilkan pola pengelompokan. Pola-pola ini dimotivasi oleh aturan produksi untuk mengidentifikasi pengulangan adegan tertentu dari seseorang atau area. Penulis menawarkan metode canggih menggunakan fitur audio dan visual tingkat rendah dan tinggi. Setelah pembagian menjadi adegan, mereka membuat beberapa Grafik Transisi Adegan untuk setiap jenis fitur. Mereka melakukan penggabungan probabilistik dari semua grafik untuk sampai pada pembagian akhir. Penulis mengukur kemiripan bidikan dengan menimbang jarak antara histogram bidikan yang dinormalisasi, dan jarak temporal antara pusat bidikan. Mereka menggunakan pengelompokan spektral untuk mengelompokkan bidikan bersama, dan menetapkan transisi adegan antara bidikan yang berdekatan yang termasuk dalam kelompok yang berbeda.

Pengelompokan Temporal

Sangat bermanfaat untuk mengeksplorasi literatur di luar bidang deteksi adegan video untuk memahami pendekatan lain yang mungkin untuk masalah yang sangat umum yaitu mempartisi elemen dengan batasan temporal. Dalam bidang analisis deret waktu, tugas umum adalah analisis titik perubahan. Penulis mencoba mendeteksi perubahan dalam aliran titik data online. Mereka menggunakan rasio diskriminan kernel untuk menguji kemungkinan perubahan, dan memperkirakan lokasi titik perubahan tersebut jika ada.

Sistem Linier Dinamis Beralih terkadang digunakan untuk memodelkan serangkaian pengamatan, masing-masing termasuk dalam salah satu kelompok keadaan. Dengan menggunakan formulasi ini, seseorang dapat mempelajari segmentasi data untuk memahami struktur mendasar dari sampel. Meskipun metode-metode ini berdasar, metode-metode tersebut kekurangan elemen penting untuk deteksi adegan: metode-metode tersebut gagal membandingkan elemen-elemen yang berjauhan satu sama lain. Dalam sebuah adegan video, terkadang bijaksana untuk secara khusus mengandalkan fitur-fitur yang serupa tetapi ditempatkan di ujung adegan yang berbeda. Hanya formulasi yang melihat rangkaian secara keseluruhan dan mengukur jarak fitur lengkap dari awal hingga akhir yang dapat berharap untuk melakukan segmentasi yang benar pada contoh-contoh yang sulit.

5.3 PENGELOMPOKAN SEKUENSIAL OPTIMAL

Sekarang penulis menyajikan metode penulis untuk deteksi adegan video menggunakan pengelompokan sekuensial optimal. Formulasi ini direpresentasikan sebagai metode pengelompokan fitur sekuensial umum untuk memperluas penerapannya ke berbagai set fitur dan domain masalah. Dalam masalah deteksi adegan video penulis, fitur-fiturnya dapat berupa, misalnya, histogram warna yang mewakili setiap bidikan.

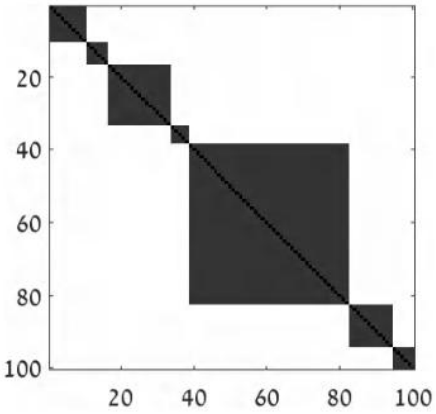
Notasi dan Definisi

Mari kita definisikan informasi yang diambil dari sebuah video sebagai serangkaian N vektor fitur dengan panjang m: $\{x_1, \ldots, x_N\} \mid x_i \in R^m$. Ini dapat mewakili, misalnya, histogram HSV yang dinormalisasi dari bingkai kunci yang mewakili sebuah bidikan, histogram rata-rata dari semua bingkai dalam setiap bidikan, atau fitur audio yang diekstrak di lokasi yang terdeteksi. Lihat sub bab berikutnya untuk detail mengenai penerapan fitur yang berbeda. Mari kita definisikan metrik jarak $D: R^m \times R^m \to R$, di mana D memenuhi semua kondisi metrik jarak matematis dan memberikan indikasi ketidakmiripan antara vektor fitur. Tujuannya adalah untuk mengelompokkan instance fitur yang serupa ke dalam klaster. Alih-alih mengelompokkan fitur-fitur ini tanpa memperhatikan urutan sekuensialnya atau memberi bobot pada metode pengelompokan berdasarkan jarak temporal (seperti yang biasanya dilakukan dalam metode lain), kita melakukan pembagian inheren dengan formulasi berikut:

Estimasi jumlah adegan yang terdiri dari video, K. Dengan K yang diberikan, kita harus menemukan serangkaian indeks yang meningkat secara monoton yang dilambangkan dengan $\bar{t} = \{t_1, \ldots, t_K\} | t_i \in N$ untuk merepresentasikan pembagian ke dalam adegan. t_i adalah indeks fitur terakhir yang termasuk dalam adegan i, sehingga $\bar{t}$ menandai titik akhir dari semua bagian. Sifat peningkatan monoton diberlakukan untuk memastikan bahwa setiap adegan diberi jumlah fitur yang bukan nol. Kita menetapkan $t_K = N$ untuk memberlakukan cakupan semua fitur yang terdeteksi, dan selain itu kita dapat mendefinisikan $t_0 = 0$ untuk menyederhanakan notasi matematika. Kita menyatakan dengan $\Omega^K \subseteq N^K$ himpunan semua kemungkinan deret dengan panjang K dengan batasan yang dijelaskan di atas.

Intuisi

Penting untuk memahami penalaran dan intuisi sebelum memasuki pembahasan berikutnya, jadi di bagian ini kita menyajikan visualisasi metode tersebut. Saat menghitung D antara semua N fitur, kita dapat memasukkan nilai-nilai tersebut ke dalam matriks: $D(j, j') = D(x_j, x_{j'})$. Kita perlu memilih representasi fitur dan metrik jarak yang memberikan nilai rendah untuk fitur dari adegan yang sama, dan nilai tinggi untuk fitur yang termasuk dalam adegan yang berbeda. Dalam hal ini, kita dapat berasumsi bahwa D akan terlihat mirip dengan penggambaran pada Gambar 5.1. Blok pada diagonal mewakili adegan di mana kelompok fitur berurutan memiliki nilai jarak kecil di antara mereka tetapi jarak yang besar ke fitur lain dalam video. Diagonal utama memiliki nilai jarak nol karena $D(x_j, x_j) = 0$.



Gambar 5.1. Contoh matriks ideal D. Piksel yang lebih terang mewakili nilai jarak yang lebih besar.

Contoh ideal ini menggambarkan bahwa jika kita memilih fungsi tujuan yang secara eksklusif menjumlahkan nilai dari dalam blok pada diagonal — ini dapat menguntungkan. Nilai optimal dari fungsi tujuan kemungkinan besar akan tercapai ketika pembagian ditempatkan sedemikian rupa sehingga fungsi tujuan menjumlahkan blok dengan jarak fitur yang rendah.

Fungsi objektif

Dengan metrik jarak yang disebutkan di atas dan dengan mempertimbangkan definisi pembagian adegan, sekarang kita ingin mendefinisikan fungsi objektif $H_D^K: \Omega^{\mathrm{K}} \rightarrow R$, yang memberikan skor pada setiap kemungkinan pembagian sehubungan dengan D dan K. Penulis menyajikan fungsi objektif yang berbentuk

$$H_D^K(\bar{t}) = \sum_{i=1}^{K} \left\{ \sum_{j=t_{i-1}+1}^{t_i} \sum_{j'=t_{i-1}+1}^{t_i} D\left(x_j, x_{j'}\right) / S_i \right\} \tag{5.1}$$

Fungsi objektif ini menjumlahkan jarak intra-grup untuk semua fitur yang dikelompokkan ke dalam adegan i dan menjumlahkan jarak grup ini untuk semua adegan, di mana S_i adalah faktor pembobot opsional (misalnya, untuk menggabungkan prior spesifik domain, jika ada).

Untuk proses optimasi di bawah ini, kita melakukan generalisasi dengan menggunakan superskrip tambahan $H_D^{n,k}$ untuk menunjukkan bahwa kita hanya menggunakan subset fitur dari n hingga N. Dalam hal ini, domain diubah menjadi $\Omega^{n,k}$, yang merupakan himpunan semua kemungkinan deret yang meningkat secara monoton dengan panjang k yang dimulai tidak lebih rendah dari n dan nilai akhirnya adalah N. Ketika H muncul hanya dengan satu superskrip, implikasinya adalah $n = 1$.

Optimasi fungsi objektif

Sekarang kita perlu menemukan $\bar{t}$ yang meminimalkan H_D^K. Karena formulasi ini bukan definisi pengelompokan yang umum, solusi pengelompokan klasik tidak dapat diterapkan. Penulis mengusulkan skema pemrograman dinamis untuk mencapai solusi optimal $\bar{t}^*$ secara

efisien. Berikut ini, kita menggunakan $S_i = 1$ untuk penyederhanaan. Sekarang kita akan menjelaskan tahapan metode pemrograman dinamis:

Mari kita definisikan tabel biaya $C(n,k)$ sebagai nilai fungsi tujuan optimal dari $H_D^{n,k}$. Nilai fungsi tujuan optimal dari keseluruhan masalah adalah $C(1,K)$. Karena nilai fungsi tujuan saja tidak mengandung informasi mengenai lokasi pembagian, kita definisikan $J(n,k)$ sebagai t_1 dari submasalah pembagian $N-n+1$ fitur menjadi k adegan. Untuk mengulangi, $J(n,k)$ menentukan di mana sebaiknya batas berikutnya ditempatkan. Kita menginisialisasi tabel dengan

$$C(n,1) = \sum_{j=n}^{N} \sum_{j'=n}^{N} D(x_j, x_{j'}) \tag{5.2}$$

Dan

$$J(n,1) = N \tag{5.3}$$

karena membagi fitur yang tersisa menjadi satu bagian hanya terdiri dari satu solusi yang mungkin $C(n,1) = H_D^{n,1}(\bar{t} = t_1 = N)$.

Dengan memanfaatkan linearitas jumlah luar H_D^K dalam (5.1), sisa tabel didefinisikan secara rekursif untuk setiap $1 < k \leq K$:

$$C(n,k) = \min_i \left\{ \sum_{j=n}^{i} \sum_{j'=n}^{i} D\left(x_j, x_{j'}\right) + C(i+1, k-1) \right\} \tag{5.4}$$

$$J(n,1) = \operatorname{argmin}_i \left\{ \sum_{j=n}^{i} \sum_{j'=n}^{i} D\left(x_j, x_{j'}\right) + C(i+1, k-1) \right\} \tag{5.5}$$

Tabel-tabel tersebut harus disusun secara berurutan dengan k mulai dari 2 hingga K. Batas-batas adegan dapat direproduksi dengan menjalankan i secara berurutan dari 1 hingga K dengan rumus berikut:

$$t_i^* = J(t_{i-1} + 1, K - i + 1) \tag{5.6}$$

Untuk efisiensi lebih lanjut, kita dapat menghitung jumlah perantara secara terpisah. Kita membuat tabel Σ berukuran $N \times N$ yang berisi jumlah-jumlah tersebut.

$$\Sigma(i,i') = \sum_{j=n}^{i} \sum_{j'=n}^{i} D\left(x_j, x_{j'}\right) \tag{5.7}$$

untuk referensi cepat. Tabel ini tidak harus dibuat secara langsung, tetapi dapat dihitung sendiri menggunakan pemrograman dinamis: Kita menginisialisasi $\Sigma(i,i) = D(x_i, x_i) = 0$ dan kemudian secara rekursif mendefinisikan sisa tabel,

$$\Sigma(i,i') = \Sigma(i-1,i') + \Sigma(i,i'-1) - \Sigma(i-1,i'-1) + D(x_i, x_{i'}) + D(x_{i'}, x_i) \qquad (5.8)$$

yang memungkinkan Σ dibangun di sepanjang diagonal dari diagonal utama dan dalam urutan menaik.

5.4 KOMPLEKSITAS KOMPUTASI

Jika masalah pembagian didekati tanpa skema pemrograman dinamis yang telah penulis kembangkan, tetapi dengan cara brute-force, jumlah kemungkinan pembagian mencapai jumlah yang sangat besar, yang diberikan oleh

$$\binom{N-1}{K-1} = \frac{(N-1)!}{(K-1)!\,(N-K)!} \qquad (5.9)$$

yang tidak praktis untuk dihitung untuk masalah dunia nyata.

Di sisi lain, dengan menggunakan metode penulis, dua tabel berukuran $K \times N$ dibangun. Untuk setiap elemen dalam tabel, fungsi minimum mempertimbangkan paling banyak N nilai menggunakan tabel Σ. Konstruksi tabel Σ dapat diabaikan untuk ini dan kita sampai pada kompleksitas waktu $O(K \cdot N^2)$.

Menentukan jumlah adegan K

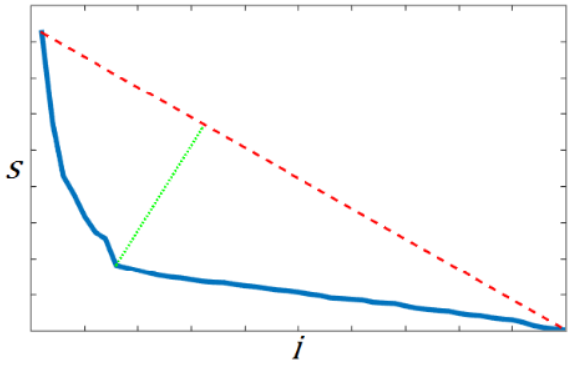
Mengidentifikasi jumlah klaster K adalah salah satu masalah yang lebih sulit dalam algoritma pengelompokan apa pun. Metode yang patut diperhatikan adalah statistik gap, yang membandingkan peningkatan jarak pengelompokan seiring bertambahnya K antara data yang ada dan distribusi nol (yaitu, kumpulan data buatan yang terdistribusi secara seragam). Namun, dalam formulasi unik penulis, penulis dapat memanfaatkan matriks D dan melakukan apa yang penulis sebut sebagai log elbow dari nilai singular matriks. Hal ini mirip dengan upaya untuk membuat aproksimasi matriks peringkat rendah yang masuk akal dalam formulasi ini, karena dalam arti yang sangat luas semua baris matriks yang termasuk dalam blok yang sama dapat dianggap kurang lebih bergantung secara linier. Mengidentifikasi jumlah klaster adalah masalah umum dalam pengelompokan data. Dalam uraian penulis di atas, penulis berasumsi bahwa K diketahui. Di sini kita membahas bagaimana dapat diperkirakan, dan penulis menyajikan pendekatan baru. Karena matriks D menyerupai matriks ketidakmiripan dalam masalah pengelompokan klasik, dimungkinkan untuk menggunakan sejumlah metode untuk memprediksi K yang diambil dari bidang pengelompokan data.

Termotivasi oleh visualisasi masalah yang disebutkan di Bagian sebelumnya, penulis mengembangkan lebih lanjut metode baru untuk menentukan K yang unik untuk formulasi kami. Masalah pengelompokan yang tidak memerlukan pengelompokan berurutan tidak dapat memanfaatkan struktur matriks D yang mudah dalam masalah ini. Seperti yang terlihat pada

matriks ideal D pada Gambar 5.1, karena ketergantungan linier antara beberapa baris, peringkat matriks kira-kira sama dengan jumlah blok pada diagonal. Dalam praktiknya, ketika terdapat noise, melakukan aproksimasi peringkat rendah dari D akan meninggalkan struktur blok dan menghilangkan artefak noise. Hal ini memungkinkan kita untuk memperkirakan K menggunakan Dekomposisi Nilai Singular (SVD), alat yang umum digunakan untuk aproksimasi peringkat rendah.

Visualisasi diberikan pada Gambar 5.2. Jika kita menyatakan vektor s sebagai m nilai dalam grafik log dari nilai singular, maka vektor dua dimensi dari awal s ke titik pada indeks i pada s adalah $I_i \triangleq [i, s_i]^T$. Diagonal pada grafik adalah $H \triangleq [m-1, s_m - s_1]^T$, dan titik siku diperoleh dengan:

$$elbow\_index = \underset{i}{\text{argmax}} \left\{ I_i - \frac{I_i^T H}{\|H\|} \cdot \frac{H}{\|H\|} \right\} \tag{5.10}$$



Gambar 5.2. Mendeteksi K menggunakan siku pada kurva log nilai singular s.

5.5 DETEKSI ADEGAN VIDEO TERBATAS

Seperti yang disebutkan di atas, ketersediaan dataset evaluasi di bidang deteksi adegan video terbatas. Untuk membantu mengatasi kurangnya tolok ukur yang tersedia, komprehensif, dan objektif, dataset Deteksi Adegan Video Terbuka (OVSD) berdasarkan video terbuka, yang memungkinkan penelitian dan penggunaan akademis dan industri. Untuk menilai kinerja deteksi adegan video, penulis menggunakan ukuran Cakupan C dan Overflow O yang diterima secara luas. Rata-rata harmonik dari C dan 1—O, yang dilambangkan dengan F, mewakili skor tunggal untuk kualitas deteksi adegan.

Penulis mengevaluasi metode dan implementasi pada dataset OVSD. Selain itu, penulis membandingkan hasil menggunakan executable yang disediakan oleh penulis, dengan batasan berikut: Karena executable yang disediakan bekerja pada video hingga 10 menit, ketika dijalankan pada `Sintel' film dipersingkat satu adegan, dan `Valkaama' dibagi menjadi 10 bagian, dengan hasil yang dirata-ratakan. Tabel 5.1 menunjukkan hasilnya, dengan skor F tertinggi untuk setiap video dicetak tebal. Mengenai implementasi, penulis memilih nilai σ sedemikian rupa untuk memaksimalkan kinerja pada dataset ini.



Gambar 5.3. Dua video dari dataset OVSD yang disajikan. Bingkai representatif dipilih sebagai bingkai tengah dari setiap adegan seperti yang dideteksi oleh metode kami.

Tabel 5.1. Hasil.

Nama Video	*Our*			[5]			[7]		
	F	C	O	F	C	O	F	C	O
ED	**0.69**	0.63	0.23	0.56	0.69	0.52	0.55	0.73	0.56
BBB	**0.69**	0.70	0.33	0.49	0.77	0.64	0.46	0.65	0.65
Sintel	**0.66**	0.55	0.17	0.59	0.62	0.43	0.51	0.81	0.63
ToS	0.62	0.57	0.32	**0.75**	0.92	0.37	0.23	0.88	0.87
CL	**0.53**	0.47	0.40	0.45	0.43	0.53	0.07	0.85	0.97
Valkaama	**0.73**	0.96	0.42	**0.73**[a]	0.88[a]	0.32[a]	0.16	0.94	0.91

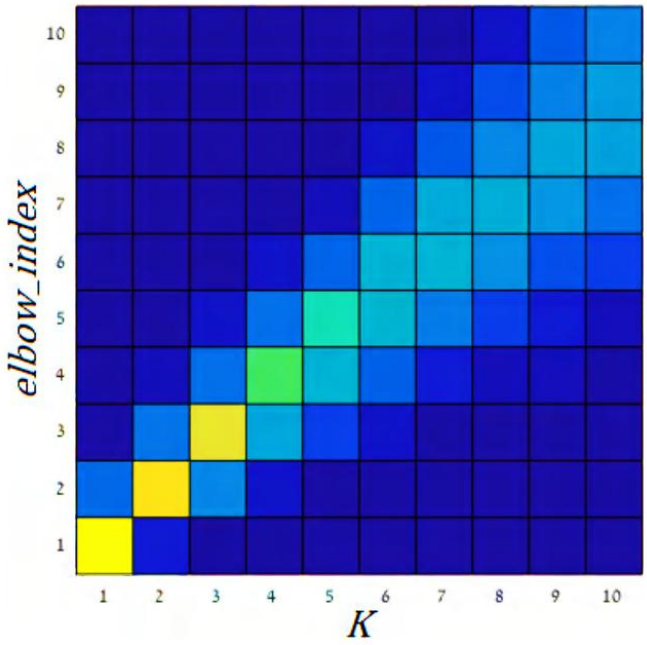
Seperti yang ditunjukkan pada Tabel 5.1, metode penulis jelas mengungguli metode-metode terkini. Selain peningkatan objektif pada skor F, metode penulis juga memiliki keunggulan karena tidak memerlukan parameter, dan tidak seperti metode lain, tidak memerlukan pelatihan atau fine-tuning untuk dataset tertentu. Gambar 5.3 menunjukkan visualisasi pembagian, di mana satu frame dari setiap adegan disajikan untuk mewakili deteksi adegan.

Akurasi Indeks Siku

Untuk mengevaluasi akurasi metode penulis dalam memperkirakan K, penulis melakukan simulasi otomatis. Penulis membuat matriks D yang dihasilkan secara acak, dengan K dipilih secara acak antara 1 dan 10, dan N dipilih sebagai $10 \cdot \alpha \cdot K$, di mana α juga dipilih secara acak antara 1 dan 10. Ini memberi kita variabilitas antara jumlah adegan dan jumlah bidikan. Penulis menghasilkan matriks D acak dengan nilai yang diambil secara seragam dari interval [0, 1]. Nilai pada diagonal blok dikuadratkan dan nilai lainnya diakarkan, sehingga menyisakan nilai rendah pada diagonal blok dan nilai tinggi di luarnya.

Penulis memperkirakan K dengan metode log-siku SVD penulis, dan membandingkannya dengan nilai K yang sebenarnya. Pada Gambar 5.4, penulis menyajikan grafik indeks siku yang diperkirakan versus nilai K sebenarnya. Hasilnya menunjukkan korelasi

yang kuat di sepanjang diagonal dengan deviasi yang sedikit meningkat seiring dengan meningkatnya K.



Gambar 5.4. Kotak yang lebih terang menunjukkan probabilitas yang lebih tinggi untuk memperkirakan indeks siku tertentu untuk K tertentu.

5.6 MENGGUNAKAN BERBAGAI MODALITAS

Pada bagian ini, penulis merinci penerapan metode penulis terkait dengan sejumlah modalitas yang berbeda. Asumsi penulis di atas adalah bahwa fitur dan metrik jarak dipilih untuk menciptakan struktur diagonal blok. Hal ini dicapai dengan memilih elemen yang menggambarkan video sedemikian rupa sehingga metrik jarak akan secara konsisten mengembalikan nilai rendah untuk fitur yang termasuk dalam adegan yang sama, dan nilai tinggi sebaliknya.

Perilaku ini tidak selalu mudah dicapai, dan seringkali bergantung pada konten. Misalnya, dapat diasumsikan bahwa video yang seluruhnya terdiri dari wawancara studio antara beberapa orang kemungkinan besar tidak mungkin untuk disegmentasi hanya berdasarkan informasi visual atau akustik. Dalam hal ini, informasi mengenai transkripsi ucapan perlu tersedia, sehingga tugasnya mirip dengan segmentasi topik dalam dokumen. Di sisi lain, beberapa video bahkan mungkin tidak mengandung kata-kata yang diucapkan, dan informasi visual atau akustik akan menjadi faktor penentu yang kuat untuk pembagian. Pada subbagian berikut, penulis memberikan beberapa detail mengenai sejumlah fitur berbeda yang telah penulis uji, dan penulis membahas detail penerapannya.

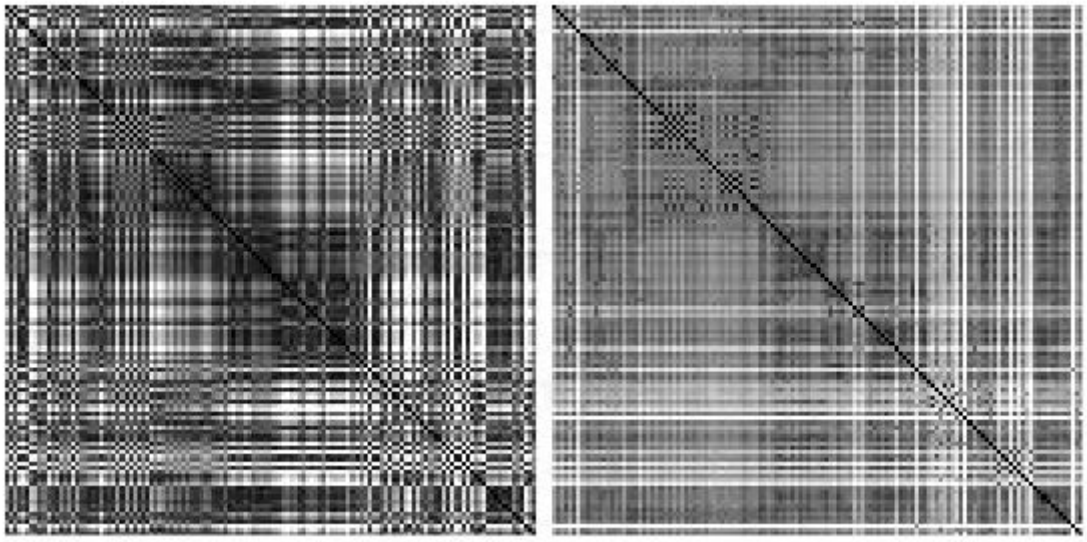
Fitur Visual

Fitur visual adalah informasi yang paling umum digunakan terkait video. Biasanya konten visual beragam dan cukup kaya untuk memungkinkan pemahaman tingkat tinggi tentang konten tersebut. Di media sosial, seringkali video dirancang untuk diputar tanpa audio

sehingga memungkinkan pengguna untuk menonton tanpa mengganggu orang di sekitarnya, dan tanpa perlu menggunakan headphone.

Histogram warna adalah salah satu cara utama fitur visual dimanfaatkan. Dengan melihat distribusi warna dalam sebuah bidikan, kita dapat memperoleh representasi paling dasar dari konten visual yang relatif kuat terhadap detail yang lebih kecil dalam bingkai. Hasil pada Bagian sebelumnya diperoleh dengan mengekstrak histogram RGB 512 dimensi (8 bin untuk setiap pita warna) untuk setiap bingkai dalam sebuah bidikan, menjumlahkannya, dan menormalkannya untuk mendapatkan representasi tunggal untuk setiap bidikan. Pada kesempatan lain selama pekerjaan penulis, penulis hanya mengambil tiga bingkai representatif dalam sebuah bidikan (dengan jarak yang seragam di seluruh bidikan) dan merata-ratakannya. Hal ini dilakukan terutama untuk menghemat waktu eksekusi selama pengujian karena ekstraksi fitur adalah pemboros waktu utama dalam proses kami. Histogram HSV juga terkadang digunakan, namun penulis tidak mencatat perubahan perilaku yang substansial.

Sebagai metrik jarak, penulis memilih jarak Bhattacharyya dari delapan jarak berbeda yang penulis uji. Dari pengujian, penulis melihat bahwa secara keseluruhan, jarak Bhattacharyya memberikan pembedaan terbaik antara fitur dari adegan yang sama versus fitur dari adegan yang berbeda. Hal ini menyebabkan matriks D yang dihasilkan memiliki struktur diagonal blok yang lebih menonjol, seperti yang dapat dilihat pada Gambar 5.5.



Gambar 5.5. Matriks D untuk fitur visual yang sama yang dibuat dengan metrik jarak yang berbeda — jarak Bhattacharyya di sebelah kiri versus jarak Euclidean di sebelah kanan.

Fitur Akustik

Untuk banyak video seperti film, iklan, konten promosi, dan berita, informasi akustik dapat sangat bermanfaat untuk deteksi adegan. Penulis menggunakan istilah akustik untuk menekankan bahwa elemen yang lebih luas dari aliran audio adalah tempat informasi untuk pembagian lebih mungkin berada. Audio, berbeda dengan video, dianggap sebagai sinyal yang jauh lebih "berisik", sehingga sangat bermasalah untuk dianalisis. Terkadang audio dipotong

menjadi fonetik kecil seperti konstanta keras atau suara glutaral dan kemudian diproses pada lapisan yang lebih tinggi mengingat klasifikasi yang lebih rendah. Namun, dalam kasus penulis, penulis ingin menggunakan data mentah untuk mengklasifikasikan kesamaan, dan diperlukan gambaran yang jauh lebih luas.

Penulis menguji dengan Koefisien Cepstral Frekuensi Mel (MFCC) standar sebagai fitur, tetapi berbeda dengan pengenalan ucapan, penulis mengambil jendela temporal yang besar (2 detik) untuk mengabaikan kebisingan tingkat rendah dan mendapatkan pemahaman yang lebih umum tentang sifat akustik dalam segmen 2 detik tertentu. Sebagai metrik jarak, penulis menggunakan jarak Euclidean antara instance fitur MFCC. Penting untuk dicatat bahwa pembagian ke adegan dalam kasus seperti itu akan diberikan pada kekasaran hingga 2 detik. Jika pembagian yang lebih tepat diperlukan, penyempurnaan titik pembagian dapat dilakukan pada dua segmen yang mengelilingi titik pembagian menggunakan alat segmentasi audio yang tidak memenuhi syarat dalam menemukan Transisi adegan tingkat tinggi skala besar, tetapi bekerja dengan baik untuk deteksi perubahan lokal.

Fitur transkripsi ucapan

Ketika mencoba menggunakan transkripsi ucapan sebagai fitur representatif untuk deteksi adegan, ada dua masalah yang perlu diatasi. Pertama, bagaimana menekankan kata-kata penting yang diucapkan, dibandingkan dengan membiarkan kata-kata umum memandu pembagian yang mungkin akan menghasilkan hasil yang tidak bermakna. Kedua, bagaimana membiarkan representasi fitur dan metrik jarak menghasilkan hasil yang diinginkan. Ini menimbulkan masalah yang lebih sulit daripada untuk fitur visual atau akustik karena keberadaan atau pengecualian satu kata terkadang bisa sangat substansial. Kita dapat berasumsi bahwa ketika berurusan dengan transkripsi ucapan, fitur-fitur tersebut kemungkinan besar akan menunjukkan perilaku diskrit.

Metode yang paling umum untuk mengatasi poin pertama di atas adalah dengan menggunakan *Term-Frequency Inverse-Document-Frequency* (TF IDF) di mana kata-kata diberi bobot berdasarkan kemunculannya dalam dokumen dan diimbangi dengan keunikannya dalam korpus dokumen. Pendekatan ini valid untuk masalah ini, namun penulis lebih memilih untuk membuat metode ekstraksi fitur penulis sedemikian rupa sehingga Buatlah agar mandiri dan tidak bergantung pada korpus yang dikumpulkan sebelumnya.

Untuk memungkinkan metode penulis bertindak sebagai unit mandiri yang tidak terlatih, penulis menghitung faktor pembobotan hanya berdasarkan frekuensi kemunculan kata secara terbalik. Ini berarti memberi bobot pada sebuah kata secara terbalik dengan jumlah kemunculannya dalam video. Di satu sisi, ini memungkinkan penulis untuk memberikan bobot kecil pada kata-kata yang sangat umum (seperti 'a', 'in', 'the') sesuai keinginan, namun juga memberi bobot pada kata-kata yang mungkin sering muncul karena merupakan elemen utama dari sebuah adegan. Hal ini juga mungkin tidak berhasil dalam memberikan bobot rendah pada kata-kata yang kurang umum dan karenanya tidak mungkin muncul berkali-kali dalam satu video. Yang memotivasi penulis adalah aspek mandiri seperti yang disebutkan di atas, dan fakta bahwa jika sebuah kata penting muncul berkali-kali dalam video, maka kemungkinan besar itu adalah kata yang mendefinisikan seluruh video dan tidak akan menjadi

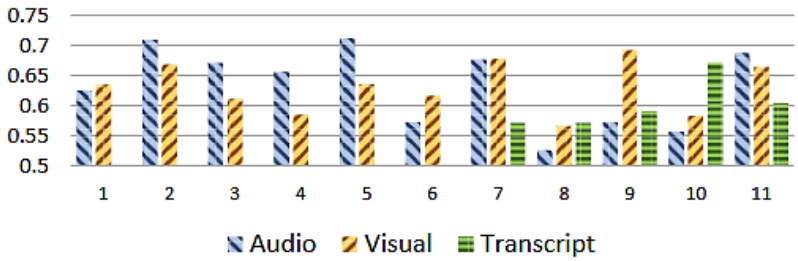
elemen kunci ketika mencoba deteksi adegan video di mana kata-kata pembeda adalah kata-kata yang penting.

Untuk mengatasi poin kedua di atas, penulis menerapkan metodologi yang sama dengan variasi jarak uji Chi:

$$D(x_i, x_j) = \sqrt{\sum_{i=1}^{m} \frac{\left(x_{i,l} - x_{j,l}\right)^2}{x_{i,l} + x_{j,l}}} \tag{5.11}$$

Motivasi di balik penggunaan metrik ini adalah bahwa besarnya skor fitur harus memengaruhi jarak lebih dari sekadar perbedaan antara kedua skor tersebut. Dengan cara ini, nilai rendah tidak berkontribusi secara substansial baik untuk meningkatkan skor jarak maupun untuk menunjukkan kesamaan — karena kesamaan kemunculan kata-kata penghenti, misalnya, seharusnya bukan indikasi ke arah tertentu.

Penulis menyajikan beberapa hasil pengujian yang dilakukan pada sejumlah video internal ketika mencoba pembagian menggunakan berbagai jenis fitur dari berbagai modalitas. Penulis mencatat bahwa meskipun formulasi yang dirinci di atas digunakan untuk melakukan pembagian, jumlah adegan sebenarnya dimasukkan ke dalam algoritma alih-alih melakukan estimasi.



Gambar 5.6. Skor F untuk sebelas video menggunakan berbagai modalitas.

Tabel 5.2. Hasil dari berbagai modalitas.

Video #	Visual	Audio	Speech Transciption
1	0.63	0.64	–
2	0.71	0.67	–
3	0.67	0.61	–
4	0.66	0.59	–
5	0.71	0.64	–
6	0.57	0.62	–
7	0.68	0.68	0.57
8	0.53	0.57	0.57
9	0.57	0.69	0.59
10	0.56	0.58	0.67
11	0.69	0.66	0.60

Penulis melakukan pengujian dengan cara ini untuk mencoba mengisolasi proses penilaian kelayakan fitur. Dalam kondisi ideal, penulis ingin memeriksa apakah fitur yang diuji dapat membuat matriks jarak D yang memungkinkan pembagian yang menguntungkan untuk diperoleh.

Pada Gambar 5.6 dan Tabel 5.2, penulis menyajikan hasil skor F deteksi adegan video pada sebelas video internal menggunakan beberapa modalitas sebagai fitur. Masih ada kekurangan serius konten yang tersedia dengan pembagian kebenaran dasar yang tidak bias. Video dalam kumpulan uji ini sayangnya tidak dapat penulis rilis, namun penulis dapat merinci bahwa video tersebut berisi konten terkait bisnis dan berdurasi sekitar 10–15 menit.

Pembagian kebenaran dasar ada karena video terdiri dari segmen yang masing-masing membahas aspek tertentu sehingga penulis dapat mengidentifikasi pembagian yang benar secara manual. Transkripsi ucapan hanya tersedia pada video 7 hingga 11 karena untuk video lainnya teknologi ucapan-ke-teks yang tersedia bagi penulis tidak mendukung bahasa lisan.

Secara umum, penulis melihat hasil yang relatif baik di semua modalitas yang berbeda, yang menyiratkan bahwa implementasi penulis berhasil. Namun, satu hal penting yang perlu diperhatikan adalah perbedaan besar yang mungkin dimiliki modalitas tertentu pada video tertentu. Dengan mengambil video 5, 9, dan 10, kita melihat bahwa untuk masing-masing video, modalitas yang berbeda menonjol secara substansial dalam hal kinerja dibandingkan dengan yang lain.

Hal ini membuat kita percaya bahwa sangat bermanfaat untuk mengembangkan kemampuan algoritmik untuk mendeteksi dengan benar modalitas mana yang harus digunakan dalam sebuah video. Selain itu, jika fusi modalitas yang cerdas dapat dilakukan, maka deteksi adegan video multi-modal mungkin dapat memanfaatkan semua data dalam video dan mengeksploitasi masing-masing dalam kasus yang dibutuhkan.

5.7 PENGELOMPOKAN TEMPORAL ADEGAN VIDEO BERBASIS FITUR ARBITRER

Dalam pada bab ini ini, penulis menyajikan metode untuk pengelompokan temporal adegan menggunakan serangkaian fitur arbitrer yang dihitung dari video. Penulis menunjukkan bagaimana formulasi unik penulis dan solusi bebas parameter dari masalah ini dapat menghasilkan solusi yang efisien menggunakan pemrograman dinamis. Hasil eksperimen penulis dengan jelas menunjukkan efektivitas metode yang diusulkan, yang mengungguli metode terkini untuk deteksi adegan. Selain itu, penulis menunjukkan kepraktisan dan penerapan berbagai jenis fitur untuk masalah ini. Pekerjaan di masa mendatang dapat melibatkan pemahaman dari konten jenis fitur mana yang paling bermanfaat untuk digunakan dalam kasus tertentu.

BAB 6
AI BERBASIS PENGETAHUAN UNTUK ANALISIS PROSES BISNIS

Untuk keputusan operasional yang kritis (misalnya, layanan konsultasi), penjelasan dan hasil yang dapat diinterpretasikan dari sistem Kecerdasan Buatan (AI) yang canggih menjadi semakin penting. Grafik pengetahuan memiliki model semantik yang mengintegrasikan sumber informasi heterogen dan merepresentasikan elemen pengetahuan dalam bentuk yang dapat dibaca mesin. Integrasi grafik pengetahuan dan metode pembelajaran mesin mewakili bentuk baru dari sistem cerdas hibrida yang saling menguntungkan. Penelitian penulis bertujuan untuk sistem yang dapat dijelaskan dengan arsitektur grafik pengetahuan spesifik yang menghasilkan hasil yang dapat dipahami manusia bahkan ketika tidak ada pakar domain yang sesuai.

Dengan latar belakang ini, interpretasi pendekatan AI yang dapat dijelaskan berbasis grafik pengetahuan untuk analisis proses bisnis difokuskan. Penulis merancang kerangka interpretasi, menunjukkan bagaimana model yang dapat diinterpretasikan dihasilkan oleh studi kasus tunggal dan mengevaluasi penerapan pendekatan penulis melalui berbagai wawancara ahli. Jalur hasil pada kelemahan dan langkah-langkah perbaikan yang terkait dengan proses bisnis digunakan untuk menghasilkan pohon keputusan stokastik, yang meningkatkan interpretasi hasil. Hal ini dapat mengarah pada layanan mandiri konsultasi yang menarik bagi klien atau diterapkan sebagai alat untuk mempercepat proyek konsultasi klasik.

6.1 XAI BERBASIS KNOWLEDGE GRAPH UNTUK ANALISIS PROSES BISNIS

Demokratisasi sistem Kecerdasan Buatan (AI) secara bersamaan telah menciptakan permintaan akan model yang memberikan penjelasan kepada pengguna tentang perilaku sistem, mekanisme kerjanya, dan hasil yang dihasilkan. Kecerdasan Buatan yang Dapat Dijelaskan (XAI) berkaitan dengan teknologi pembelajaran mesin yang menghasilkan model dan keputusan yang dapat dijelaskan dengan lebih baik sambil mempertahankan atau meningkatkan kinerja, serta merancang dan menyediakan sistem AI yang mudah dipahami dan dapat dipercaya oleh pengguna manusia.

Literatur semakin banyak membahas pendekatan neuro-simbolik, di mana metode simbolik AI berbasis pengetahuan (misalnya, teknologi Web Semantik untuk representasi pengetahuan) adalah alat untuk menjelaskan sistem dengan pendekatan AI subsymbolik dan intensif data. Grafik pengetahuan sebagai jaringan semantik besar menyediakan representasi pengetahuan multi-relasional dan terstruktur grafik yang berisi informasi spesifik konteks dalam bentuk tautan berkode antara entitas, propertinya, dan memungkinkan penalaran kausal. Pembelajaran mesin mampu menyempurnakan grafik pengetahuan untuk memberikan jawaban yang bermakna terhadap pertanyaan melalui inferensi dan penalaran, misalnya. dalam sistem rekomendasi, untuk pengambilan keputusan atau prediksi.

Dibandingkan dengan pendekatan XAI yang terbatas pada atribut yang terdapat dalam data, menggunakan korelasi atau kesamaan numerik dalam proses penjelasan (misalnya LIME,

daftar aturan Bayesian), knowledge graph mempertimbangkan pengetahuan latar belakang yang relevan dengan konteks, memungkinkan kesamaan semantik dan mengkodekan pengetahuan dalam bentuk pernyataan terstruktur seperti bahasa untuk interpretasi yang mendekatkan hasil analisis dengan pemahaman manusia.

Skalabilitas metode pembelajaran mesin dalam knowledge graph juga dapat merugikan interpretasi hasil dan ekspresivitas kemampuan penalaran. Dengan meningkatnya kompleksitas (misalnya aturan keputusan yang luas atau kumpulan data berdimensi tinggi), model white-box tersebut juga membutuhkan metode untuk menginterpretasikan hasilnya. Untuk mengatasi penurunan interpretasi di bawah penskalaan, algoritma penambangan digunakan untuk melatih model yang dapat diinterpretasikan dengan melakukan klasifikasi node. Namun, ada risiko bahwa semantik yang awalnya dikodekan hilang dan tidak tersedia untuk interpretasi hasil.

Hasil AI yang mudah dipahami manusia menjadi semakin relevan dalam proses pengambilan keputusan operasional yang kritis. Dalam perjalanan transformasi digitalnya sendiri, industri konsultasi semakin bergantung pada teknologi digital untuk mendukung dan bahkan sebagian menggantikan pekerjaan konsultan. Solusi yang sepenuhnya otomatis dan layanan mandiri konsultasi menjadi mungkin di area aplikasi tertentu.

Di sini, aplikasi XAI dapat diterapkan secara bermanfaat dan berpotensi menciptakan keunggulan kompetitif. Perusahaan konsultan memiliki tujuan untuk secara interaktif mendefinisikan, menyusun, dan menganalisis masalah bisnis dengan klien mereka, mengembangkan solusi untuk masalah, dan menerapkannya jika diperlukan. Sebagai bagian penting dari rantai nilai dalam banyak proyek konsultasi, analisis proses mengumpulkan data dari berbagai sumber untuk mendapatkan pemahaman yang konsisten tentang proses yang ada dan mengidentifikasi kelemahan dan potensi peningkatan. Penambangan proses merupakan tren teknologi populer untuk analisis proses otomatis, yang menganalisis data yang ada dalam sistem TI dan mengungkap informasi proses tersembunyi berdasarkan jejak digital. Namun demikian, penambangan proses selalu membutuhkan keahlian teknis yang diperlukan untuk mengevaluasi dan menafsirkan hasilnya, dan tidak berlaku untuk proses yang tidak atau hanya sebagian didukung oleh TI.

Dengan tujuan analisis proses otomatis yang menginterpretasikan berbagai sumber data dan memperkayanya dengan penalaran logis, penulis menggunakan arsitektur grafik pengetahuan inferensial dengan metode pembelajaran mesin dan mengembangkan pendekatan untuk analisis semantik model proses bisnis. Pendekatan berbasis Riset Ilmu Desain (DSR) penulis mengejar tujuan untuk menjawab isu-isu kompleks seperti Langkah-langkah perbaikan mana yang sesuai untuk kelemahan mana dalam proses bisnis? yang sejauh ini belum dapat dijawab sepenuhnya secara otomatis dan mendukung penjelasan berbasis pengetahuan dalam konteks proyek konsultasi.

Karena struktur internalnya pada kelas node dan edge yang terdefinisi serta algoritma deduksi generik, grafik pengetahuan cocok untuk dikembangkan menjadi sistem yang dapat dijelaskan berbasis pengetahuan (KBX). Tujuan penelitian penulis membahas pembangkitan penjelasan otomatis dari grafik pengetahuan inferensial sambil mempertahankan pengkodean

semantik, serta pencapaian interpretasi hasil analisis proses yang dapat dipahami manusia, misalnya mengapa titik lemah dan langkah-langkah perbaikan diidentifikasi secara otomatis.

6.2 AI BERBASIS PENGETAHUAN

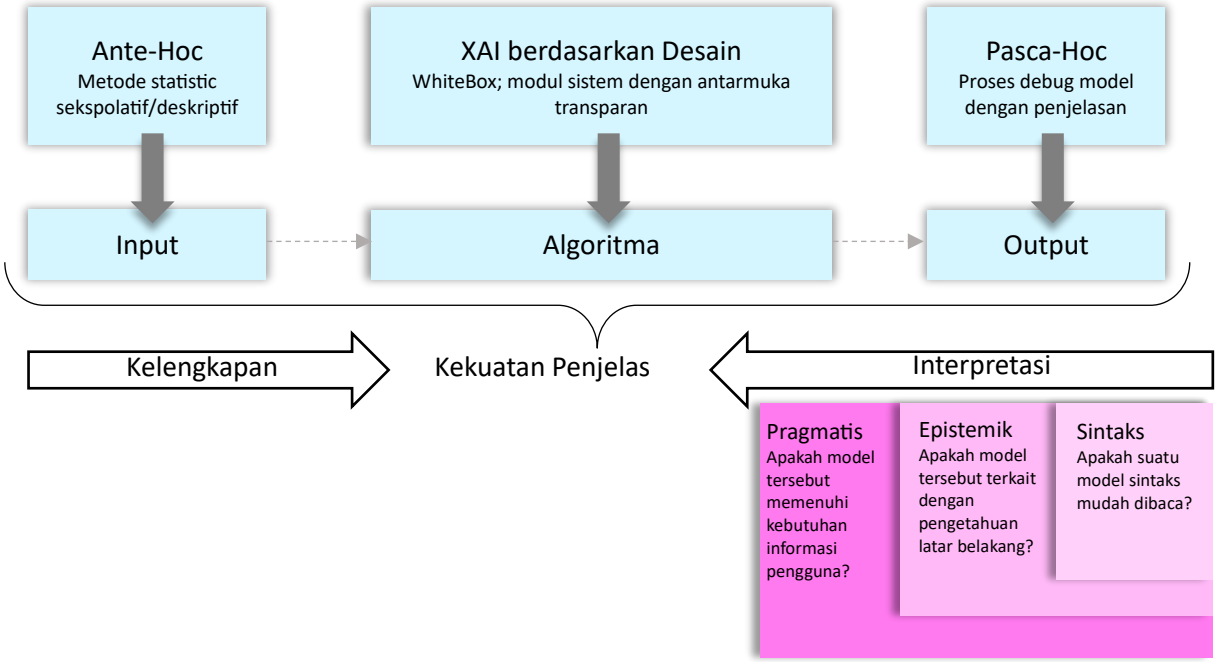
Kita mulai di bagian ini dengan mendefinisikan istilah-istilah yang relevan dan merangkum karya terkait tentang peran grafik pengetahuan untuk XAI, dengan mempertimbangkan. Penjelasan merupakan jawaban akhir untuk suatu area masalah atau pertanyaan dan bergantung pada pengetahuan, kognisi, dan interpretasi penerima. Istilah "kemampuan menjelaskan" dan "kemampuan menafsirkan" terkadang digunakan secara sinonim dalam literatur, meskipun keduanya harus dibedakan satu sama lain. Kemampuan menjelaskan mengacu pada kemampuan untuk memahami logika mekanisme kerja suatu sistem, seperti yang dibutuhkan untuk jaringan saraf. Hanya input dan output yang dianalisis, model yang mendasarinya tetap tidak tersentuh (kotak hitam). Sebaliknya, transparansi menunjukkan sejauh mana suatu sistem mengekspos informasi tentang cara kerja internalnya (algoritma) dan dapat dicapai dengan model yang secara inheren transparan (kotak putih). Interpretasi dalam konteks ini dapat digambarkan sebagai kemampuan untuk menjelaskan atau membuat makna nilai hasil dapat dipahami oleh manusia. Hal ini mendefinisikan sejauh mana seseorang mampu menggunakan informasi yang diberikan oleh sistem melalui penjelasan atau transparansi.

Tiga pendekatan XAI generik yang menghasilkan penjelasan dapat dibedakan, lihat Gambar 6.1. Pendekatan ante-hoc mencakup analisis eksplorasi kumpulan data menggunakan metode matematika-statistik. Pendekatan XAI by Design bertujuan untuk mengembangkan model yang secara inheren transparan. Dan akhirnya, penjelasan post-hoc dapat diterapkan pada model "kotak hitam" menggunakan debugging model. Di sini, ketegangan antara kelengkapan dan interpretasi penjelasan dapat dilihat.

Kelengkapan membutuhkan integrasi semua informasi yang diperlukan dan relevan untuk pengambilan keputusan. Interpretasi dapat diklasifikasikan lebih lanjut menjadi tiga tingkatan yang saling berkaitan: sintaksis (apakah model dapat dibaca), epistemik (apakah model dapat dikaitkan dengan pengetahuan latar belakang), dan interpretasi pragmatis (apakah model memenuhi kebutuhan informasi dan memiliki plausibilitas). Kekuatan penjelasan dari pendekatan XAI bergantung pada tingkat kelengkapan dan tingkat interpretasi suatu penjelasan. Dengan demikian, model yang secara inheren transparan umumnya memiliki kekuatan penjelasan yang lebih tinggi daripada model kotak hitam.

Sistem KBX mewakili sistem cerdas hibrida yang mengintegrasikan pendekatan simbolik pengetahuan terstruktur ke dalam pendekatan pembelajaran mesin. Melalui tinjauan literatur terstruktur, penulis merangkum berbagai karakteristik sistem KBX, yang disajikan dalam Tabel 6.1. Sistem KBX sebagian besar berbasis pada knowledge graph, menunjukkan pendekatan subsymbolik yang menghasilkan penjelasan berbasis pengetahuan, yang mewakili output, dan dapat ditetapkan ke domain aplikasi konkret. Berdasarkan hal ini, empat karakteristik utama dapat dibentuk, lihat kolom pertama pada Tabel 6.1.

Graf pengetahuan dalam sistem KBX dibedakan berdasarkan jenis pengetahuan, konten informasi semantik, cara pemilihan elemen pengetahuan dalam proses penjelasan, kemampuan penggunaan kembali, dan jumlah graf yang digunakan. Pengetahuan terstruktur dalam graf pengetahuan dapat berisi pengetahuan domain dan faktual dan dapat dijelaskan secara semantik oleh struktur komponen berbasis logika TBox (termino-logis, untuk deskripsi abstrak kelas dan properti objek pengetahuan) dan ABox (assersional, untuk deskripsi konkret atribut dari contoh pengetahuan). Dalam proses penjelasan, pengetahuan dipilih secara otomatis melalui tautan yang dimodelkan dalam graf, atau secara manual oleh seorang ahli yang memilih terlebih dahulu bagian-bagian graf yang relevan.



Gambar 6.1. Pendekatan XAI yang menghasilkan penjelasan

Seringkali satu graf pengetahuan, yang dapat digunakan kembali. Metode subsymbolik untuk menghasilkan penjelasan berbasis pengetahuan adalah berbasis aturan, berbasis pohon, atau bergantung pada jaringan saraf. Data tabular, gambar, atau data tekstual digunakan sebagai input untuk melakukan beberapa tugas, seperti klasifikasi, peramalan, atau regresi. Knowledge graph diintegrasikan baik untuk membuat model (ante-hoc), untuk menjelaskan model (post-embedded), atau keduanya dalam bentuk hibrida, sesuai dengan pendekatan XAI by Design.

Penjelasan yang dihasilkan diberikan secara tekstual, dalam bahasa alami, atau secara visual dalam bentuk gambar, dan dapat merujuk pada model keluaran (kategorikal), perilaku sistem (mekanistik), atau keduanya. Setiap penjelasan dapat diklasifikasikan menurut interpretasinya (lihat Gambar 6.1).

Pendekatan pembelajaran mesin yang dapat dijelaskan menggunakan knowledge graph dapat dibagi menjadi domain berikut: tugas berbasis aturan (keputusan), pengenalan gambar, rekomendasi serta tugas prediksi, dan aplikasi bahasa alami. Metode pembelajaran

mesin berbasis aturan bergantung pada pendekatan induktif berbasis logika, di mana pengetahuan latar belakang grafik dijelajahi untuk membangun klaster dan penjelasan berbasis jalur dari keluaran sistem dihasilkan (misalnya, mencari negara di mana laki-laki lebih berpendidikan). Pendekatan-pendekatan ini terutama mendukung penjelasan kategorikal dan mempertimbangkan pengetahuan ABox serta TBox.

Dalam domain pengenalan gambar dan tugas rekomendasi, di mana jaringan saraf digunakan, penjelasan sebagian besar mengacu pada perilaku model, terutama menggunakan pernyataan ABox, dan seringkali terbatas pada hubungan inferensi dalam grafik pengetahuan. Sistem rekomendasi dengan interaksi pengguna, misalnya untuk penjelasan rekomendasi film yang dipersonalisasi, mengintegrasikan grafik pengetahuan dalam pembuatan keluaran sistem. Sistem ini sebagian besar didasarkan pada grafik pengetahuan dengan pengetahuan faktual.

Tabel 6.1. Karakterisasi sistem KBX.

Karakteristik	Atribut	Nilai
Knowledge Graph (KG)	Type	Domain, Factual
	Sematik	TBox, Abox
	Selection	Manual, Automatic
	Reusability	Reusable, unique
	Number of KG	1, >1
Metode subsymbolik	Input	Data Tabular, gambar, data tekstual, lainnya
	Metode	jaringan saraf berbasis pohon
	Task	Klasifikasi, Peramalan, Regresi
	Integrasi	Ante-hoc (Membuat model), post-embedded (Untuk menjelaskan model), Hibrida
Eksplanation	Form	Data (tekstual), bahasa alami, gambar
	Type	Kategorikal (merujuk pada hasil), mekanistik (merujuk pada mekanisme yang menyebabkan suatu hasil), keduanya
	Interpretability	Sintaksis, epistemik, pragmatik
Aplikasi Domain		Tugas berbasis aturan (pengambilan keputusan), pengenalan gambar, rekomendasi, tugas prediksi, aplikasi bahasa alami.

Tugas bahasa alami atau prediksi menghasilkan penjelasan kategorikal dan mekanistik berdasarkan kombinasi pernyataan ABox dan TBox. Untuk sistem tanya jawab, grafik pengetahuan dibuat setelah disematkan berdasarkan dialog yang dihasilkan serta diintegrasikan ke dalam model selama menjawab pertanyaan pengetahuan. Dengan syarat pengetahuan domain dan faktual diintegrasikan, interpretasi sintaksis, epistemik, dan pragmatis dapat dihasilkan yang paling sesuai dengan kebutuhan informasi pengguna manusia. Keterbatasan mendasar dari penjelasan berdasarkan grafik pengetahuan adalah penggunaan elemen pengetahuan yang terbatas menurut prinsip tetangga terdekat.

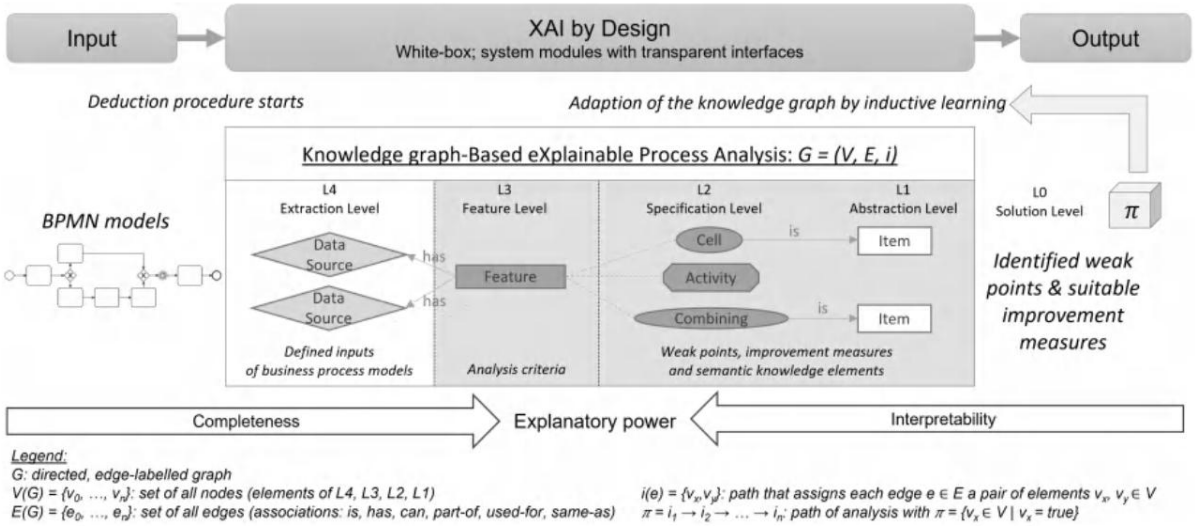
Dengan cara ini, penjelasan terbatas pada contoh konkret dan propertinya, seringkali mengacu pada pernyataan ABox. Pengkodean semantik berbasis jalur hilang dan tidak tersedia

untuk interpretasi hasil. Keterbatasan lainnya adalah pemilihan manual bagian-bagian relevan dari grafik pengetahuan yang digunakan untuk menghasilkan penjelasan. Hal ini terjadi di hampir semua bidang aplikasi. Hanya dalam bahasa alami dan tugas prediksi terdapat pendekatan di mana penjelasan yang dihasilkan didasarkan pada elemen pengetahuan yang diekstrak secara otomatis melalui tautan yang diikuti dalam grafik. Penjelasan berbasis pengetahuan memerlukan antarmuka transparan yang memungkinkan penjelasan berbasis jalur dalam grafik pengetahuan dan, pengetahuan latar belakang yang relevan dalam bentuk pengetahuan faktual dan domain yang memiliki elemen pengetahuan abstrak serta konkret untuk memberikan interpretasi yang dapat dipahami manusia.

6.3 ANALISIS PROSES BERBASIS GRAFIK PENGETAHUAN

Dalam praktik konsultasi, analisis proses sering dilakukan secara manual dan kualitasnya bergantung pada keterampilan konsultan. Jika tidak ada proses yang didukung TI secara menyeluruh, konsultan sering memodelkan proses bisnis mereka secara manual menggunakan notasi yang sudah mapan. Namun, jika data proses dalam sistem TI tersedia, penambangan proses dapat digunakan untuk pertama-tama merekonstruksi proses bisnis sebagai model proses dan kemudian menghitung Indikator Kinerja Utama (KPI) yang relevan. Hal ini membantu konsultan untuk mendapatkan pemahaman yang komprehensif tentang proses dan untuk mengidentifikasi masalah dan kelemahan dalam proses yang mereka selidiki. Setelah itu, dengan menerapkan pengetahuan proses dan domain mereka, konsultan dapat menyarankan langkah-langkah peningkatan proses. Namun, tanpa konteks semantik yang diperlukan, kesimpulan tersebut tidak dapat secara otomatis diturunkan dari model proses.

Analisis Proses yang Dapat Dijelaskan Berbasis Grafik Pengetahuan (KBX-PA) mendukung identifikasi titik lemah dan langkah-langkah peningkatan yang sesuai dalam proses bisnis berdasarkan pengetahuan ahli dalam grafik pengetahuan. Sama seperti konsultan belajar dan mentransfer pengetahuan dari satu proyek ke proyek berikutnya, hasil dari satu analisis otomatis dapat dipertimbangkan dalam analisis proses selanjutnya. Informasi yang relevan untuk analisis dikumpulkan dari model proses dan digunakan untuk analisis semantik titik lemah dan derivasi langkah-langkah perbaikan yang sesuai dengan menerapkan penalaran deduktif. Model proses dari *Business Process Model and Notation* (BPMN) yang telah mapan berfungsi sebagai sumber input yang patut dicontoh (lihat Gambar 6.2), terlepas dari berbagai format input yang dapat diproses. Berikut ini, konsep dan arsitektur KBX-PA akan disajikan.



Gambar 6.2. Konsep KBX-PA.

Desain dan klasifikasi

Desain KBX-PA didasarkan pada grafik berarah, berlabel tepi, seperti ontologi dengan algoritma deduksi dan mekanisme pembelajaran induktif. Kelas elemen (node) dan asosiasi (tepi) yang terdefinisi memungkinkan penataan dan representasi elemen pengetahuan yang terstandarisasi, dengan mempertimbangkan hubungan semantik. Algoritma deduksi menjalankan mekanisme inferensi dalam grafik pengetahuan tergantung pada kelas asosiasi tertentu, yang mewakili prosedur analisis model proses bisnis. Tiga pendekatan pembelajaran induktif menyempurnakan dan memperluas grafik pengetahuan dengan setiap inferensi. Pertama, grafik pengetahuan diperluas dengan elemen pengetahuan baru dari data masukan model proses. Kedua, elemen pengetahuan konkret mewariskan karakteristiknya ke elemen pengetahuan abstrak tingkat yang lebih tinggi, dan akhirnya hasil inferensi diverifikasi oleh manusia dalam prosesnya.

Pendekatan XAI berbasis knowledge graph dapat diklasifikasikan sebagai pendekatan hybrid, XAI by Design, karena perilaku sistem transparan selama deduksi/analisis proses dan model yang dapat diinterpretasikan dibuat untuk menghasilkan penjelasan. Analisis otomatis proses bisnis penulis selalu didasarkan pada knowledge graph, yang berisi pengetahuan faktual tentang kriteria analisis, potensi kelemahan, langkah-langkah perbaikan yang sesuai, serta pengetahuan domain dari area proses yang akan dianalisis (analog dengan pengetahuan pengalaman seorang konsultan).

Dengan demikian, knowledge graph terdiri dari elemen proposisional terminologis (abstrak, TBox) dan konkret (ABox). Dasar untuk analisis dalam setiap kasus adalah knowledge graph terintegrasi yang dapat digunakan kembali. Awalnya, file XML dari model proses digunakan sebagai input. Analisis proses terdiri dari empat langkah analisis (lihat Tabel 6.2, urutan algoritma prosedur deduksi), yang dieksekusi oleh algoritma deduksi berbasis graph (sesuai dengan kelas asosiasi).

Dengan mempertimbangkan perhitungan node berbasis aturan, jawaban klasifikasi dalam hal kelemahan dan langkah-langkah perbaikan dihasilkan. Penjelasan dari output yang

dihasilkan bersifat tekstual, dalam bahasa yang hampir alami, dan merujuk pada jalur analisis yang mengarah ke hasil dan pengetahuan latar belakang semantik yang terkait dengan hasil tersebut. Jalur keputusan yang telah dilalui dapat direkonstruksi sesuai dengan elemen node yang diaktifkan dan memberikan penjelasan. Oleh karena itu, jalur hasil yang dibuat membangun titik awal untuk menghasilkan model yang dapat diinterpretasikan, yang mendukung interpretasi hasil dan merupakan fokus penelitian pada bab ini ini. Sebaliknya, kelengkapan pengetahuan bergantung pada input dan grafik pengetahuan yang mendasari analisis.

Tabel 6.2. Prosedur deduksi KBX-PA.

Pertanyaan analisis dan hasilnya (misalnya)	Urutan algoritma analisis proses
(1) *Apakah prosesnya sudah dioptimalkan?* • ya / tidak (2) *Apa saja kelemahannya?* • jeda media, • tugas manual yang ada, • pekerjaan duplikat, • kerusakan sistem TI, • dokumen analog, dll. (3) *Langkah perbaikan apa yang sesuai?* • meningkatkan koordinasi, • meningkatkan otomatisasi, • mengurangi tugas manual, dll. (4) *Alat mana yang sesuai?* • sistem manajemen alur kerja, • sistem ERP, • mesin pemesanan online, dll.	CALL (1) Isn't-it optimized process IF Result (1) == FALSE CALL (2) Kind-of weak point IF Result (2) is not null CALL (2a) Characterize CALL (2b) Parts CALL (2c) Like CALL (2d) Find CALL (3) Find improvement measure IF Result (3) is not null CALL (4) Find tool END

Model Abstrak Grafik Pengetahuan Yang Terstruktur

Arsitektur ini didasarkan pada model abstrak grafik pengetahuan yang terstruktur dalam lima lapisan: lapisan input data L4, lapisan transfer data L3, L2, dan L1 untuk pemrosesan informasi serta representasi pengetahuan, dan lapisan solusi L0 yang berisi semua elemen yang diaktifkan sebagai jalur hasil selama analisis proses. Kecuali L0, lapisan-lapisan tersebut terdiri dari berbagai tipe elemen (kelas) yang memenuhi fungsi lapisan masing-masing, lihat Gambar 6.2.

Kelas elemen Sumber Data ditetapkan pada level L4 dan digunakan untuk menghubungkan sumber data eksternal yang menjadi dasar analisis. Model proses dalam gaya BPMN berbasis XML dibaca oleh antarmuka. Selain antarmuka, elemen Sumber Data juga dapat berupa tipe pertanyaan pengguna untuk menggunakan informasi atau umpan balik terkait pengguna dalam analisis proses. Setiap elemen Sumber Data membutuhkan setidaknya satu elemen Fitur level L3. Kelas elemen Fitur digunakan untuk penggunaan atau transfer data yang tepat dari level L4 dalam format dan konteks semantik yang diperlukan untuk

mengaktifkan elemen pengetahuan level L2. Elemen fitur mewakili kriteria analisis proses, seperti jumlah tugas manual atau tugas sinonim, yang mengidentifikasi potensi kelemahan dalam model proses, seperti gangguan media atau pekerjaan yang duplikat.

Kelemahan dan langkah-langkah perbaikan dimodelkan dalam tingkat pemrosesan informasi L2 dan L1 (sebanding dengan ABox dan TBox). Tipe elemen Sel, Penggabungan, dan Aktivitas (L2), dan tipe Item dari tingkat abstraksi L1 digunakan untuk representasi informasi yang relevan dengan analisis proses. Kelas asosiasi dasar adalah, memiliki, dapat, bagian dari, digunakan untuk, dan sama dengan memungkinkan hubungan semantik antar elemen.

Untuk identifikasi konkret misalnya gangguan media di L2, yang disebut Ekspresi Hasil diperlukan untuk menghasilkan Batasan (garis putus-putus pada Gambar 6.2). Ekspresi Hasil $r(v_x)$ didefinisikan untuk aktivasi bersyarat suatu elemen, dan menyatakan rumus perhitungan dengan hasil Boolean benar (1) dan salah (0). Dengan cara ini, elemen node L2 dan L1 dapat mengambil nilai 0 atau 1 (diaktifkan). Gangguan media dapat diidentifikasi, misalnya jika suatu proses memiliki tugas manual dan sistem TI secara bersamaan:

$$r(v_{mediaBreak}) = true \leftarrow r(v_{manualTask}) > 0 \&\& r(v_{ITSystem}) > 0.$$

Algoritma deduksi melakukan inferensi dalam grafik dan dapat dibagi menjadi algoritma abstraksi (Isn't-it, Kind-of) dan konkretisasi (Characterize, Parts, Like, Find) yang menganalisis semua hubungan semantik suatu elemen, bahkan jika hubungan tersebut tidak secara eksplisit dimodelkan dalam model proses. Tergantung pada Ekspresi Hasil yang ditentukan, algoritma tersebut mengaktifkan elemen pengetahuan dengan nilai hasil Boolean true (1). Algoritma deduksi didasarkan pada kelas asosiasi dan memicu dirinya sendiri tergantung pada elemen yang diaktifkan. Misalnya, jika asosiasi sama dengan elemen yang akan dianalisis dipicu, algoritma Like dipanggil untuk mengidentifikasi sinonim yang ada.

Prosedur analisis proses bisnis sebagai model BPMN terdiri dari empat langkah analisis utama yang dipicu oleh algoritma deduksi dan menghasilkan elemen solusi untuk setiap pertanyaan, lihat Tabel 6.2. Sesuai dengan itu, dalam urutan algoritma, setiap CALL mencakup sebuah output. Setiap analisis proses bisnis dapat mengarah pada adaptasi atau perluasan grafik pengetahuan melalui mekanisme pembelajaran induktif. Adaptasi grafik pengetahuan dapat diotomatisasi dengan mentransfer karakteristik dari konkretisasi ke abstraksi (misalnya, proses Item abstrak mewarisi elemen dokumen analog sebagai asosiasi kepemilikan dari elemen konkret proses manual), dengan mengidentifikasi dan menambahkan elemen pengetahuan baru (misalnya, sistem TI baru sebagai alat pengukuran peningkatan) atau dengan memverifikasi hasil dan mekanisme pembelajaran melalui interaksi manusia. Semakin komprehensif grafik pengetahuan dirancang, semakin detail analisis proses yang dapat dilakukan dan semakin tinggi interpretasi hasil analisis yang dapat diantisipasi.

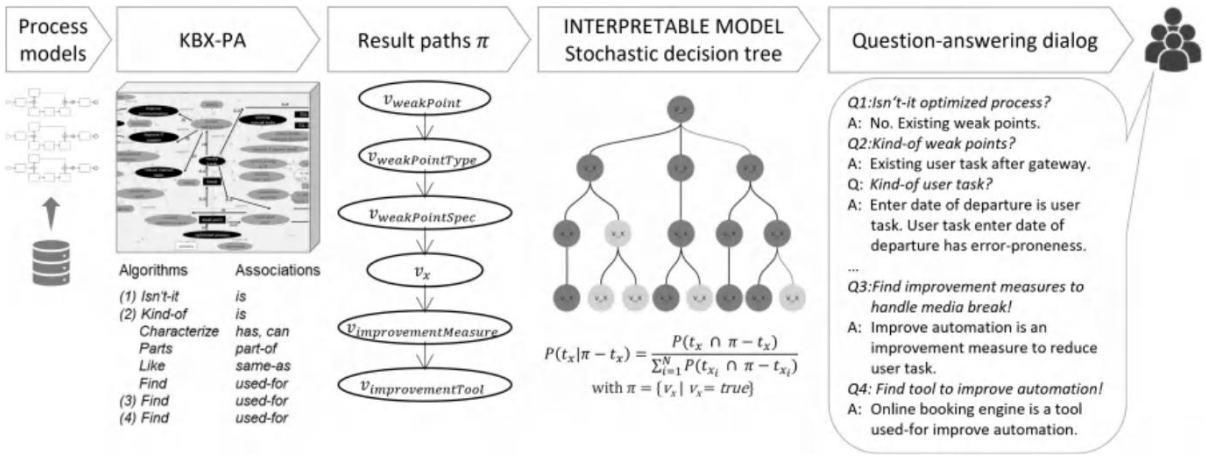
6.4 PERMODELAN DESAIN POHON KEPUTUSAN STOKASTIK

Dengan tujuan mencapai keputusan yang berorientasi pada target dalam proyek konsultasi, solusi yang diusulkan untuk optimasi proses memerlukan interpretasi yang

memadai agar dapat dibenarkan kepada klien. Gagasan awal pendekatan XAI berbasis grafik pengetahuan memenuhi konsep tersebut karena algoritma deduksi berbasis asosiasi yang mendukung rekonstruksi semua jalur traversal dalam grafik pengetahuan, membentuk dasar untuk interpretasi yang dapat dipahami manusia.

Kerangka interpretasi menghasilkan model yang dapat diinterpretasikan dalam bentuk pohon keputusan stokastik berdasarkan jalur hasil yang dipicu oleh algoritma deduksi (lihat Gambar 6.3). Seperti yang diklasifikasikan dalam Bagian sebelumnya, jenis penjelasan mengacu pada output sistem (kategorikal) dan perilaku sistem (mekanistik). Penjelasan kategorikal didasarkan pada pengetahuan latar belakang semantik tentang output yang diidentifikasi (kelemahan, langkah-langkah perbaikan), penjelasan mekanistik memberikan wawasan tentang jalur analisis yang dilalui dalam grafik untuk menghasilkan output. Perbedaan antara kedua jenis penjelasan tersebut tidak terjadi di sisi pengguna.

Selama prosedur analisis, algoritma deduksi dijalankan secara generik dalam grafik pengetahuan dalam arti bahwa algoritma tersebut ditugaskan ke kelas asosiasi yang sesuai. Algoritma tersebut saling membangun dan mempertimbangkan semua nilai hasil yang ditentukan selama analisis. Dengan demikian, semua elemen dalam grafik pengetahuan yang dilalui menjadi bagian dari jalur hasil. Jalur yang dilalui dalam grafik pengetahuan dipetakan dalam bentuk pohon keputusan stokastik. Di sini, probabilitas jalur dihitung untuk terjadinya simpul target. Probabilitas tersebut menunjukkan sejauh mana langkah-langkah perbaikan yang sesuai dapat diinterpretasikan dengan mempertimbangkan titik lemah yang teridentifikasi dari suatu proses. Karena sifat algoritma deduksi yang berbasis pertanyaan, dialog tanya jawab mewakili keluaran sistem untuk mengkomunikasikan hasil dengan cara yang mudah dipahami manusia (semua langkah dihasilkan oleh mesin).



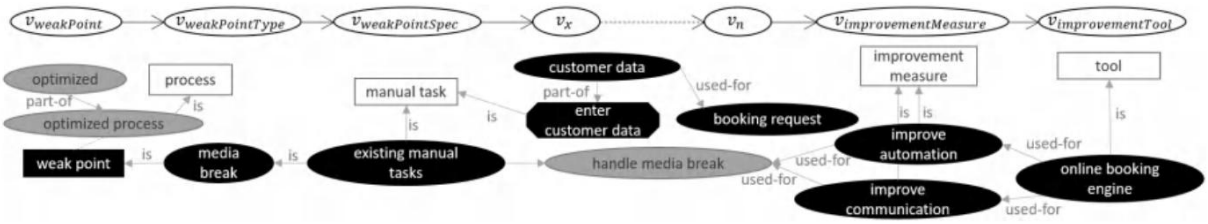
Gambar 6.3. Kerangka interpretasi untuk KBX-PA.

Pembuatan model yang dapat diinterpretasikan

Selama eksekusi algoritma deduksi, elemen node yang diaktifkan menjadi bagian dari jalur hasil. Jalur hasil dibuat di sepanjang seluruh prosedur analisis (langkah analisis 1–4, lihat Tabel 6.2) dan dapat disusun menjadi agregasi berikut, lihat cuplikan dari grafik pengetahuan dan node yang diaktifkan (ditandai dengan warna hitam) pada Gambar 6.4.

Semua node titik lemah pertama-tama dianalisis, dimulai dari tingkat abstraksi (misalnya, jeda media sebagai $v_{weakPointType}$) hingga tingkat spesifikasi (misalnya, tugas manual yang ada sebagai $v_{weakPointSpec}$). Semua node yang memiliki hubungan semantik melalui asosiasinya berfungsi untuk menggambarkan kelemahan secara lebih rinci dan juga menjadi bagian dari jalur hasil (misalnya, memasukkan data pelanggan, permintaan pemesanan sebagai v_x). Selanjutnya, node langkah-langkah perbaikan (misalnya, meningkatkan otomatisasi sebagai $v_{improvementMeasure}$) dan alat/opsi (misalnya, mesin pemesanan online sebagai $v_{improvementTool}$) dianalisis. Sejalan dengan itu, jalur hasil analisis proses memiliki bentuk generik.

Dengan cara ini, semua elemen pengetahuan yang dipertimbangkan dalam proses analisis tersedia untuk interpretasi hasil akhir (langkah/alat perbaikan). Kelas asosiasi membantu menempatkan node elemen yang diaktifkan ke dalam hubungan yang mudah dipahami manusia (misalnya, mesin pemesanan online adalah alat yang digunakan untuk meningkatkan otomatisasi dalam menangani jeda media).



Gambar 6.4. Jalur hasil yang mengikuti elemen pengetahuan yang diaktifkan (node yang disorot dengan warna hitam).

Jalur hasil yang dibuat oleh prosedur analisis memungkinkan derivasi pohon keputusan stokastik. Secara khusus, harus berupa And-Or-tree, karena tidak hanya satu elemen yang dapat bernilai benar, tetapi juga beberapa elemen. Probabilitas jalur dihitung untuk simpul target tergantung pada elemen-elemen yang diaktifkan dari jalur hasilnya. Menurut teorema Bayes, probabilitas jalur bersyarat $P(t_x \mid \pi - t_x)$ dari simpul target t_x (ukuran atau alat perbaikan) dengan syarat bahwa simpul elemen hulu v_x sebagai bagian dari jalur hasil r telah diaktifkan $(v_x = benar)$ dapat ditentukan untuk setiap titik lemah yang diberikan dan dihitung sebagai berikut:

$$P(t_x \mid \pi - t_x) = \frac{P(t_x \cap \pi - t_x)}{\sum_{i=1}^{N} P(t_{x_i} \cap \pi - t_{x_i})} \qquad (6.1)$$

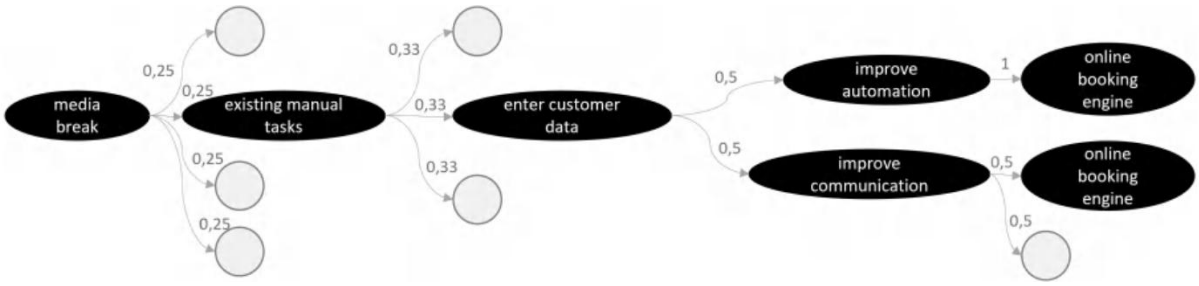
$$\text{Dengan } \pi = \{v_x \mid v_x = \text{true}\}$$

Untuk mengilustrasikan perhitungan jalur, contoh yang ditunjukkan pada Gambar 6.4 dikurangi menjadi elemen pengetahuan minimum dalam grafik, lihat Gambar 6.5. Node berwarna abu-abu muda tanpa konten mewakili elemen pengetahuan tetangga yang dilalui selama analisis proses tetapi bukan bagian dari jalur hasil karena nilai hasilnya (false). Di sini, terdapat dua jalur hasil yang merujuk pada elemen solusi yang sama. Mesin pemesanan online

sebagai alat peningkatan diidentifikasi untuk langkah-langkah peningkatan otomatisasi dan peningkatan komunikasi guna menghindari tugas manual memasukkan data pelanggan, yang diidentifikasi sebagai gangguan media.

Perhitungan probabilitas jalur bersyarat dari dua solusi yang diusulkan ditunjukkan pada Tabel 6.3. Hasilnya, terdapat probabilitas yang lebih tinggi dengan hasil mesin pemesanan online yang digunakan untuk meningkatkan otomatisasi daripada dengan hasil mesin pemesanan online yang digunakan untuk meningkatkan komunikasi. Hal ini disebabkan oleh jumlah elemen solusi yang mungkin dari langkah-langkah peningkatan. Dengan demikian, elemen solusi mesin pemesanan online saat ini adalah satu-satunya alat yang memenuhi langkah peningkatan otomatisasi dalam konteks yang diberikan, sedangkan dua elemen solusi digunakan untuk mendukung langkah peningkatan komunikasi.

Dapat disimpulkan adanya ketergantungan antara probabilitas jalur yang ditentukan dari elemen solusi dan grafik pengetahuan yang dianalisis. Jumlah elemen dan asosiasi dari grafik pengetahuan menentukan cakupan basis analisis. Dengan demikian, probabilitas jalur bergantung pada berapa banyak simpul yang terdiri dari grafik pengetahuan dan berapa banyak simpul yang bergantung secara semantik menjadi bagian dari jalur hasil selama analisis. Semakin banyak simpul dalam asosiasi semantik yang diaktifkan selama proses deduksi, semakin banyak elemen pengetahuan yang tersedia untuk menggambarkan hasilnya.



Gambar 6.5. Contoh pembangkitan model yang dapat diinterpretasikan beserta probabilitas jalurnya.

Tabel 6.3. Contoh perhitungan probabilitas jalur bersyarat.

Hasil dari titik lemah	Tugas manual yang ada
Mesin pemesanan online digunakan untuk meningkatkan otomatisasi	$\frac{0.04125}{(0.04125 + 0.020625)} = 0.66$
Mesin pemesanan online digunakan untuk meningkatkan komunikasi	$\frac{0.020625}{(0.04125 + 0.020625)} = 0.33$

Implikasinya untuk model yang dapat diinterpretasikan adalah bahwa probabilitas jalur yang lebih tinggi dari simpul target dapat dikaitkan dengan interpretasi yang lebih baik atau lebih eksplisit (epistemik dan pragmatis). Cakupan interpretasi suatu hasil bergantung pada cakupan grafik pengetahuan dan elemen pengetahuan yang diaktifkan yang menggambarkan hasil analisis. Berikut ini, penulis mendemonstrasikan penerapan pendekatan XAI berbasis

grafik pengetahuan penulis, termasuk studi kasus tunggal, dan menyajikan analisis rinci tentang model yang dapat diinterpretasikan.

6.5 JUSTIFIKASI TUJUAN DESAIN

Justifikasi Desain Terhadap Penambangan Proses

Menurut DSR, manfaat dari artefak yang dikembangkan perlu dibuktikan. Oleh karena itu, berbagai metode evaluasi ada dalam literatur, tergantung pada konteks dan tujuan evaluasi. Evaluasi ex-ante mengevaluasi desain artefak, sedangkan evaluasi ex-post mengacu pada penggunaan artefak. Penulis memilih kombinasi metode ex-ante dan ex-post dalam desain evaluasi kami.

Pertama, penulis bertujuan untuk membuktikan tujuan desain dalam domain analisis proses otomatis dan membandingkan pendekatan penulis dengan alat penambangan proses. Selanjutnya, penulis membenarkan artefak yang dikembangkan terhadap grafik pengetahuan klasik dan menunjukkan mengapa pendekatan berbasis grafik pengetahuan penulis untuk analisis proses yang dapat dijelaskan membutuhkan arsitektur grafik pengetahuan tertentu. Terakhir, studi kasus tunggal digunakan untuk mengevaluasi kegunaan artefak melalui umpan balik ahli. Dua tugas evaluasi pertama dilakukan ex-ante dengan cara berorientasi desain, sedangkan langkah terakhir dilakukan dengan cara berorientasi aplikasi menggunakan kasus penggunaan praktis.

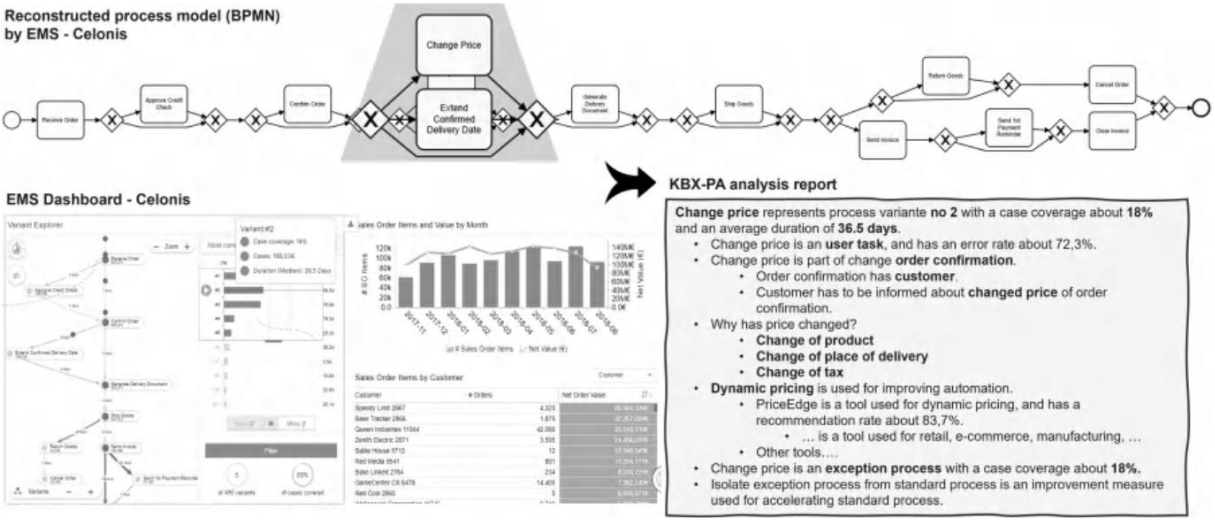
Penambangan proses adalah metode rekonstruksi, analisis, dan peningkatan proses bisnis secara otomatis berdasarkan file log dalam sistem TI, seperti yang disediakan oleh SAP, Oracle, atau Salesforce. Log kejadian harus berisi data yang diperlukan untuk penambangan proses, setidaknya ID kasus, nama aktivitas, dan stempel waktu. Dalam penelitian, penambangan proses merupakan topik yang menonjol terutama di bidang analisis data dan manajemen proses bisnis untuk menganalisis informasi tersembunyi dalam sistem TI dan untuk memajukan otomatisasi penemuan dan analisis proses. Dalam praktiknya, penambangan proses tidak diberi perhatian yang besar. Menurut survei oleh Deloitte, hanya 30% responden yang mengklasifikasikan penambangan proses sebagai relevan dalam praktik. Alasan yang mungkin untuk ini adalah kurangnya penerimaan dan daya penjelas dari hasilnya.

Prasyarat untuk penambangan proses adalah basis data log kejadian yang lengkap dan bersih, di mana model proses dapat dihasilkan. Untuk mendemonstrasikan analisis penambangan proses, penulis menggunakan Sistem Manajemen Eksekusi (EMS) dari Celonis dan proses order-to-cash yang menjadi contoh. Akibatnya, penambangan proses merekonstruksi model proses, semua varian proses di luar alur proses standar, dan menghitung banyak KPI, seperti jumlah eksekusi proses, waktu siklus proses, atau varian proses. Selain itu, informasi lebih lanjut tersedia dari dasbor hasil, seperti perubahan harga, pemasok, alamat penerima, isi dokumen pengiriman. Dalam praktiknya, hanya personel terlatih yang mampu mengevaluasi dasbor hasil. Analisis penyebab kemungkinan kelemahan dilakukan secara manual.

Model proses yang direkonstruksi melalui penambangan proses dapat digunakan dalam spesifikasi notasi BPMN sebagai masukan untuk analisis berbasis pengetahuan dengan

pendekatan KBX-PA kami. Untuk proses order-to-cash yang dibahas, grafik pengetahuan yang sesuai perlu dimodelkan sebagai prasyarat untuk analisis proses yang dapat dijelaskan. Menurut prosedur analisis (Tabel 6.2), titik lemah dari masing-masing varian proses dapat diungkap dengan cara ini, efek dapat dijelaskan melalui informasi latar belakang semantik, dan langkah-langkah serta alat yang sesuai dapat diidentifikasi untuk peningkatan proses.

Tangkapan layar pada Gambar 6.6 menunjukkan model proses yang dibuat melalui penambangan proses, termasuk kemungkinan varian (diwakili oleh gerbang eksklusif), dasbor hasil dalam EMS Celonis (sisi kiri) dan contoh laporan analisis sebagai contoh tampilan dengan penjelasan tekstual dalam bahasa alami yang dihasilkan oleh KBX-PA (sisi kanan).



Gambar 6.6. Contoh laporan analisis oleh KBX-PA menggunakan data keluaran Celonis EMS.

Selain model proses BPMN yang direkonstruksi, KPI yang ditentukan oleh penambangan proses akan digunakan untuk membuat laporan hasil oleh KBX-PA. Dengan cara ini, pendekatan KBX-PA yang dikembangkan terkait dengan analisis penambangan proses, memungkinkan laporan hasil dalam cara yang mudah dibaca manusia dan meningkatkan interpretasi hasil analisis penambangan proses.

Untuk saat ini, ini adalah versi awal dari modul penjelasan untuk model penambangan proses. Saat ini penulis sedang mengevaluasi manfaat kombinasi tersebut secara teoritis maupun praktis.

Justifikasi desain terhadap ConceptNet

Pada bagian ini, penulis meninjau fitur unik dari pendekatan desain KBX-PA penulis dibandingkan dengan arsitektur grafik pengetahuan akses terbuka klasik, seperti ConceptNet. Arsitektur pendekatan KBXAI penulis didasarkan pada grafik pengetahuan analitik dengan tipe node dan edge yang ditentukan (lihat Bagian 3). Node pada level L2 dan L1 (Sel, Aktivitas, Kombinasi, Item) dapat diberi rumus perhitungan (Ekspresi Hasil) untuk mendapatkan nilai Boolean benar (1) atau salah (0) tergantung pada input.

Algoritma deduksi yang telah ditentukan berjalan melalui grafik pengetahuan tergantung pada kelas asosiasi yang dimodelkan. Pada setiap node yang dilalui, diperiksa apakah rumus perhitungan yang mendasarinya mengarah pada kelanjutan penelusuran jalur

atau node tetangga yang dihitung. Pendekatan pembelajaran induktif memperluas dan menyempurnakan grafik pengetahuan dengan setiap analisis (input elemen pengetahuan baru, perubahan rumus perhitungan, penetapan bobot).

Mirip dengan bagaimana nilai ambang batas dalam jaringan saraf berubah tergantung pada nilai input tertentu, Ekspresi Hasil node juga beradaptasi tergantung pada analisis yang dilakukan dan menghasilkan nilai hasil yang berubah, tetapi tanpa karakter kotak hitam. Semua jalur hasil analisis bersifat transparan dan memungkinkan derivasi model yang dapat diinterpretasikan, yang berfungsi untuk menjelaskan hasil yang dihasilkan (titik lemah dan langkah-langkah perbaikan). Dalam konteks ini, pendekatan sistem KBX yang penulis sajikan mewakili grafik pengetahuan terintegrasi dengan contoh pengetahuan yang dapat dihitung dan terhubung secara semantik dalam bentuk basis pengetahuan yang dapat dianalisis.

Knowledge graph ConceptNet adalah knowledge graph berbasis crowdsourcing gratis dengan data terbuka yang terhubung untuk representasi pengetahuan umum. Untuk tujuan ini, bagian dari basis pengetahuan atau knowledge graph lain, seperti Wikipedia atau DBPedia serta WordNet, diimpor dan tautan eksternal ke basis pengetahuan tersebut dibuat. Dengan cara ini, ConceptNet mencakup pengetahuan umum yang luas dan menghubungkan banyak entitas dengan konteks semantik. Yang tidak ditawarkan ConceptNet dalam bentuk akses terbukanya adalah konten spesifik domain, atau representasi makna konkret dari kata-kata. Dengan demikian, ConceptNet juga tidak menampilkan elemen pengetahuan manajemen proses bisnis atau analisis proses.

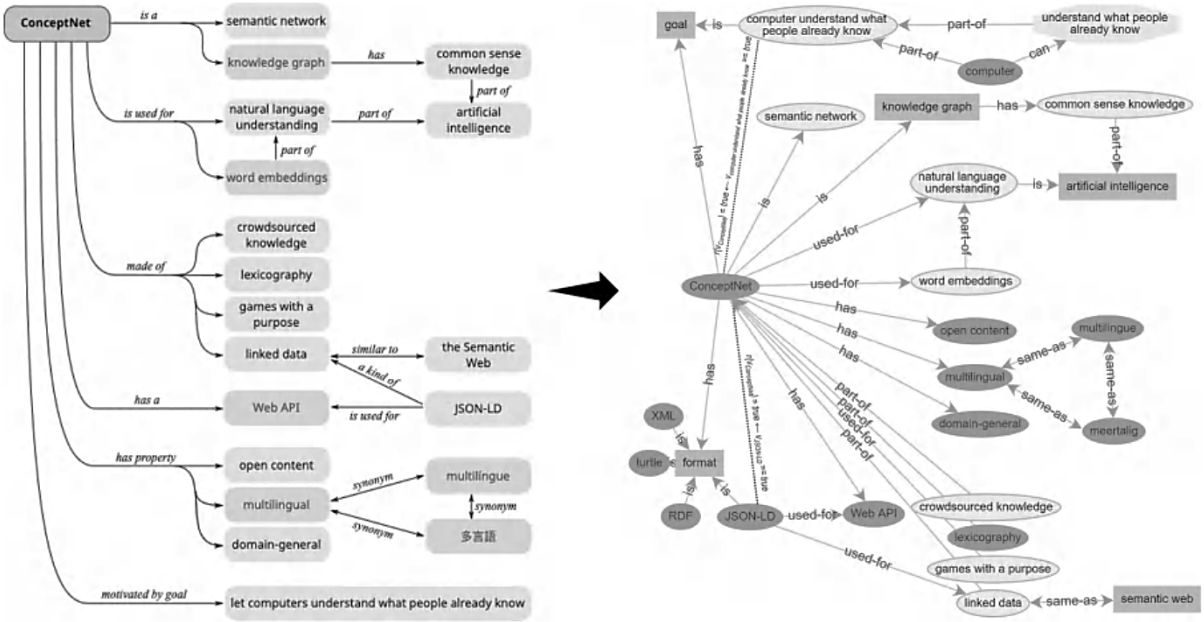
Dalam arsitektur representasi pengetahuannya, ConceptNet paling mirip dengan pendekatan KBX-PA kami. ConceptNet menawarkan serangkaian tipe edge/asosiasi yang digunakan secara generik untuk representasi pengetahuan. Tipe edge yang disesuaikan dan ditentukan pengguna, seperti yang digunakan dalam banyak ontologi, tidak diizinkan dalam knowledge graph ConceptNet. Tipe edge yang digunakan mirip dengan kelas asosiasi dari arsitektur knowledge graph penulis, seperti yang tercantum di bawah ini (ConceptNet/KBX-PA): misalnya IsA/is, PartOf/part-of, HasA/has, UsedFor/used-for, Synonym/same-as, atau CapableOf/can).

Gambar 6.7 menyajikan contoh kecil dan menonjol dari grafik ConceptNet dan representasinya menggunakan arsitektur grafik pengetahuan penulis sebagai perbandingan. Perbedaan antara kedua representasi pengetahuan terletak pada pemetaan umum dari instance pengetahuan dalam grafik pengetahuan ConceptNet.

Dalam pendekatan berbasis grafik pengetahuan penulis, batasan memungkinkan untuk memodelkan elemen pengetahuan dalam kaitannya dengan konteks semantik konkret. Dengan demikian, batasan antara node ConceptNet dan JSON-LD (direpresentasikan dalam bentuk Ekspresi Hasil $r(v_{\text{ConceptNet}}) = \text{true} \leftarrow v_{\text{JSON-LD}} = \text{true}$) digunakan untuk merepresentasikan konteks semantik bahwa format JSON-LD cocok untuk merepresentasikan data tertaut dalam grafik pengetahuan ConceptNet, bahkan jika grafik pengetahuan tersebut mencakup lebih banyak format, seperti XML, RDF, atau turtle.

Dengan cara ini, elemen pengetahuan diaktifkan tergantung pada konteks semantik tertentu yang dipicu oleh input. Karena kesamaan arsitektur representasi pengetahuan, masuk

akal untuk mengimpor bagian dari grafik pengetahuan ConceptNet untuk perluasan basis pengetahuan kita guna mengaitkan pengetahuan analisis proses spesifik dengan elemen pengetahuan umum. Hal ini memungkinkan pengetahuan latar belakang semantik tambahan tentang kelemahan untuk dibangun, yang dapat digunakan untuk interpretasi hasil. Untuk tujuan ini, jenis asosiasi yang digunakan dapat dipetakan dengan tepat dan dapat dibedakan antara elemen konkret (L2) dan elemen abstrak (L1).



Gambar 6.7. Contoh grafik pengetahuan ConceptNet dibandingkan dengan arsitektur grafik pengetahuan kami.

Penerapan Grafik Pengetahuan Untuk Analisis Bisnis

Studi kasus digunakan sebagai metode observasi untuk penerapan dan evaluasi artefak yang dibangun dalam konteks nyata. Penulis menggunakan untuk mendemonstrasikan kerangka kerja pembuatan model yang dapat diinterpretasikan dari grafik pengetahuan dan untuk mengevaluasi penerapan interpretasi otomatis pertama oleh pemilik proses, yang sangat memahami proses dan kelemahan yang dipertimbangkan, dan selanjutnya oleh enam wawancara ahli independen.

Dengan menggunakan desain kasus tunggal, pemilihan kasus harus mengacu pada kasus penggunaan klasik dalam konsultasi, terutama selama analisis proses bisnis. Penulis menggunakan prototipe berbasis C# penulis untuk analisis BPMN terkait identifikasi titik lemah, seperti hambatan media, aktivitas manual, dan langkah-langkah perbaikan yang sesuai.

Kasus penggunaan ini didasarkan pada sebuah perusahaan Jerman di industri manufaktur dengan lebih dari 500 karyawan. Perusahaan tersebut mencari cara untuk menganalisis potensi peningkatan proses internal tanpa menyewa perusahaan konsultan dan dengan upaya yang moderat, agar dapat memutuskan langkah-langkah apa yang dapat diimplementasikan untuk optimasi. Prototipe ini berpotensi sebagai teknologi konsultasi swalayan untuk memungkinkan analisis proses yang didorong oleh klien dengan interpretasi

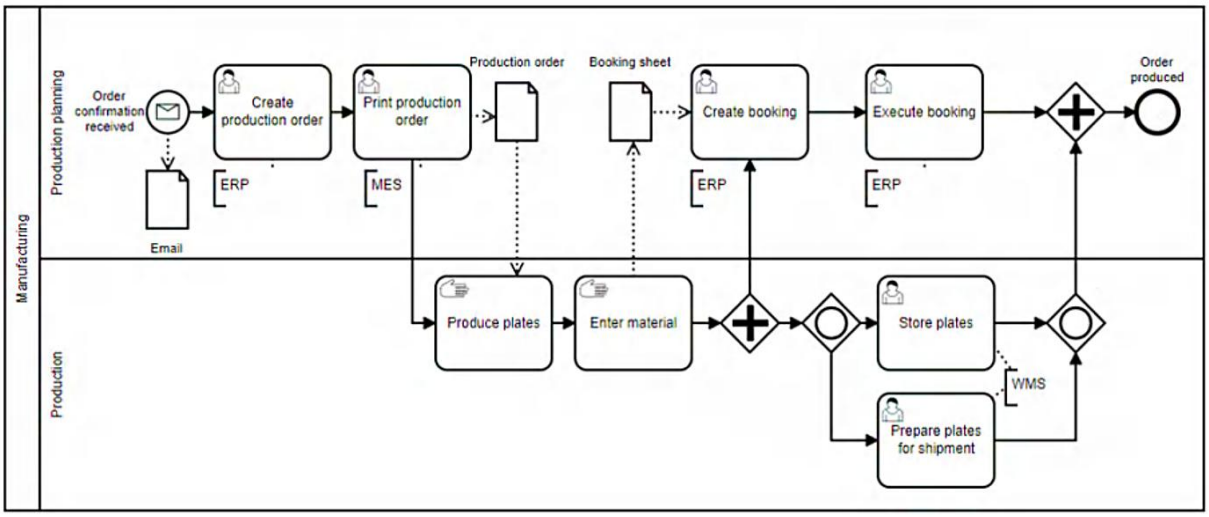
hasil yang mudah dipahami manusia. Dasar analisis penulis adalah bagian proses dari produksi sesuai pesanan, di mana dua unit organisasi perencanaan produksi dan produksi terlibat. Gambar 6.8 mewakili bagian proses yang relevan sebagai model BPMN.

Pembuatan Interpretasi

Model BPMN dari produksi berdasarkan pesanan dianalisis sehubungan dengan jeda media yang ada menggunakan grafik pengetahuan yang dibuat awalnya dengan elemen dari domain aplikasi masing-masing. Sesuai dengan prosedur analisis dari Bagian sebelumnya, langkah-langkah yang dijelaskan dilakukan. Untuk mendemonstrasikan proses deduksi dan derivasi jalur hasil, bagian dari langkah (2) dari prosedur analisis diperiksa lebih detail.

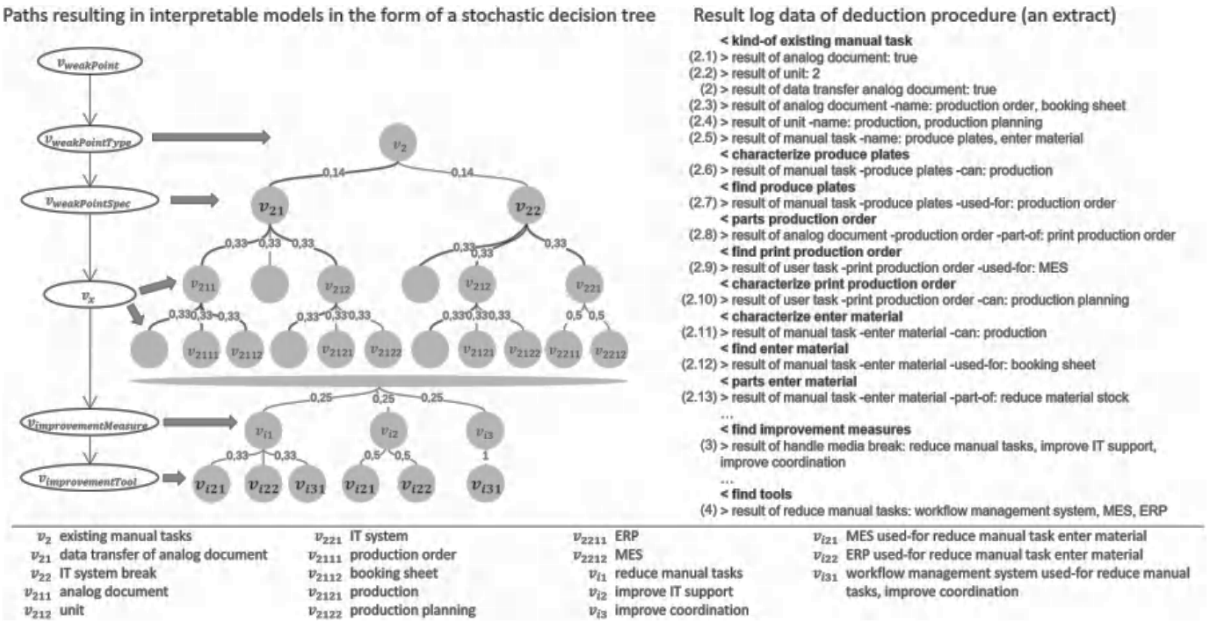
Pertanyaan analisis pertama (Apakah prosesnya sudah dioptimalkan?) dinegasikan. Ada beberapa titik lemah dalam model BPMN (misalnya, membuat pesanan produksi sebagai tugas pengguna, memasukkan material sebagai tugas manual), yang diperiksa lebih cermat pada langkah analisis (2).

Elemen tugas manual yang ada membentuk simpul masuk dari langkah analisis kedua. Algoritma Kind-of menganalisis semua titik lemah tugas manual yang direpresentasikan dalam grafik pengetahuan dan apakah titik lemah tersebut dapat diidentifikasi dalam model proses yang ada. Dengan demikian, semua elemen pengetahuan yang berdekatan yang menentukan contoh individual dari titik lemah tugas manual diperiksa untuk nilai hasil masing-masing. Gambar 6.9 menunjukkan cuplikan dari pohon keputusan yang dihasilkan serta cuplikan dari log hasil.



Gambar 6.8. Model proses produksi berdasarkan pesanan dari studi kasus dalam format BPMN.

Identifikasi transfer data dokumen analog sebagai spesifikasi dari jenis titik lemah tugas manual yang ada difokuskan (v_2 pada Gambar 6.9). Ekspresi Hasil dari elemen ini menyatakan bahwa transfer data dokumen analog adalah benar (aktif) jika ada dokumen analog yang digunakan oleh lebih dari satu unit organisasi: $r(v_{21}) = \text{true} \leftarrow r(v_{211}) == \text{true} \;\&\&\; r(v_{212}) > 1$, lihat (2.1) dan (2.2) pada Gambar 6.9.



Gambar 6.9. Contoh pembuatan model yang dapat diinterpretasikan.

Selanjutnya, elemen pesanan produksi dan lembar pemesanan diidentifikasi sebagai dokumen analog (2.3), dan produksi serta perencanaan produksi dideteksi sebagai dua unit organisasi (2.4). Pada saat yang sama, label dari dua tugas manual dicatat (2.5) dan transfer data elemen dokumen analog menjadi benar (2). Oleh karena itu, elemen ini menjadi bagian dari pohon keputusan sebagai vweakPointSpec (v21, lihat Gambar 6.9), serta elemen konteks semantik dokumen analog ($v_{211}, v_{2111}, v_{2112}$) dan unit ($v_{212}, v_{2121}, v_{2122}$). Saat mengidentifikasi tugas, asosiasi yang ada ke unit dan dokumen analog yang diidentifikasi dikenali. Namun, hal ini tidak jelas bagi pengguna secara detail selama analisis proses, tetapi dapat dilacak karena struktur grafik pengetahuan.

Berdasarkan asosiasi can antara tugas manual yang diidentifikasi menghasilkan pelat dan unit produksi, algoritma Karakterisasi dipicu. Hasil (2.6) dapat ditentukan. Selanjutnya, asosiasi yang digunakan dihubungkan dengan tugas manual, yang memanggil algoritma Find dan menentukan pesanan produksi sebagai dokumen analog yang digunakan untuk tugas manual menghasilkan pelat (2.7). Setelah itu, analisis dokumen analog pesanan produksi dilakukan oleh algoritma Parts, karena di sini asosiasi yang keluar dan bagiannya diidentifikasi (2.8). Pada saat yang sama, tautan aktivitas manual memasukkan material, lembar pemesanan dokumen analog, dan unit organisasi produksi diidentifikasi (2.11)–(2.13). Di sini, inferensi semantik dapat dibuat ke elemen pengetahuan yang ada dalam grafik pengetahuan, yang tidak secara eksplisit termasuk dalam model input (BPMN). Input material menyebabkan pengurangan stok material (2.13). Setelah semua elemen grafik pengetahuan dianalisis, algoritma Find menentukan langkah-langkah dan alat perbaikan berdasarkan kelemahan yang diidentifikasi sebagai bagian dari langkah analisis (3) dan (4).

Interpretasi dibentuk secara kronologis di sepanjang jalur hasil. Kelas asosiasi yang ada di antara elemen pengetahuan membantu mengubah elemen pengetahuan yang terletak pada jalur hasil menjadi keluaran yang dapat dipahami manusia (lihat Tabel 6.4, dicetak tebal).

Analisis otomatis model BPMN menghasilkan berbagai langkah perbaikan. Probabilitas jalur bersyarat dari hasil yang diidentifikasi, yaitu sistem manajemen alur kerja (v_{i31}), MES (v_{i21}), dan ERP (v_{i22}) terkait spesifikasi titik lemah yang dipilih, yaitu transfer data dokumen analog (v_{21}) dan kerusakan sistem TI (v_{22}), serta interpretasinya, membentuk keluaran sistem (lihat Tabel 6.4).

Tabel 6.4. Probabilitas jalur dan interpretasi hasil yang dipilih.

	Titik Lemah	
Hasil	Transfer data dokumen analog V21	Sistem TI Merusak V22
v_{i31}	0.2481	0.2515
	• Mengurangi tugas manual digunakan untuk menangani kerusakan media. • Tugas manual memasukkan materi adalah jeda media • Urutan produksi dokumen analog digunakan untuk tugas manual pembuatan pelat. • Urutan produksi dokumen analog merupakan bagian dari urutan produksi cetak tugas pengguna.	
v_{i21}	0.375	0.4048
	• Mengurangi stok material adalah bagian dari memasukkan material • Pesanan produksi dokumen analog merupakan bagian dari transfer data dokumen analog. • Urutan produksi dokumen analog merupakan bagian dari urutan produksi cetak tugas pengguna. • Sistem TI MES digunakan untuk pemesanan produksi cetak. • Urutan produksi dokumen analog digunakan untuk tugas manual pembuatan pelat. • Memasukkan material merupakan bagian dari tugas manual untuk memproduksi piring.	
v_{i22}	0.3975	0.4048
	• Lembar pencatatan dokumen analog merupakan bagian dari transfer data dokumen analog. • Lembar pencatatan dokumen analog merupakan bagian dari tugas manual memasukkan material. • Lembar pemesanan dokumen analog digunakan untuk tugas pengguna membuat pemesanan. • Sistem IT ERP digunakan untuk tugas pengguna, yaitu membuat pemesanan. • Tugas pengguna membuat pemesanan adalah bagian dari tugas manual memasukkan materi.	

Perbandingan kelemahan yang teridentifikasi menunjukkan bahwa probabilitas hasil v_{i21}, v_{i22}, dan v_{i31} lebih tinggi untuk kelemahan kerusakan sistem TI (v_{22}) daripada untuk kelemahan transfer data dokumen analog (v_{21}). Hal ini dapat dikaitkan dengan jumlah jalur yang dieksekusi selama deduksi dan elemen yang diaktifkan yang terkandung, yang mengarah pada identifikasi kelemahan. Dalam cuplikan pohon keputusan pada Gambar 6.9, dapat dilihat bahwa meskipun kedua titik lemah memiliki jumlah elemen penerus yang diaktifkan yang

sama (enam), titik lemah v_{22} memiliki lebih sedikit elemen penerus yang terhubung secara semantik (delapan) daripada titik lemah v_{21} dengan sembilan node penerus, yang menyiratkan probabilitas yang lebih tinggi untuk v_{22} daripada untuk v_{21}.

Elemen solusi sistem manajemen alur kerja yang digunakan untuk mengurangi tugas manual (v_{i31}) adalah elemen solusi abstrak, yang mengacu pada lapisan penalaran L1, secara independen dari elemen spesifik proses dalam grafik pengetahuan. Karena lebih sedikit node pengetahuan yang bergantung secara semantik di lapisan ini, maka lebih sedikit jalur yang diperlukan selama deduksi. Hal ini menghasilkan hasil yang kurang spesifik, yang, karena abstraksinya, memiliki probabilitas lebih rendah daripada dua elemen solusi lainnya (lihat Tabel 6.4).

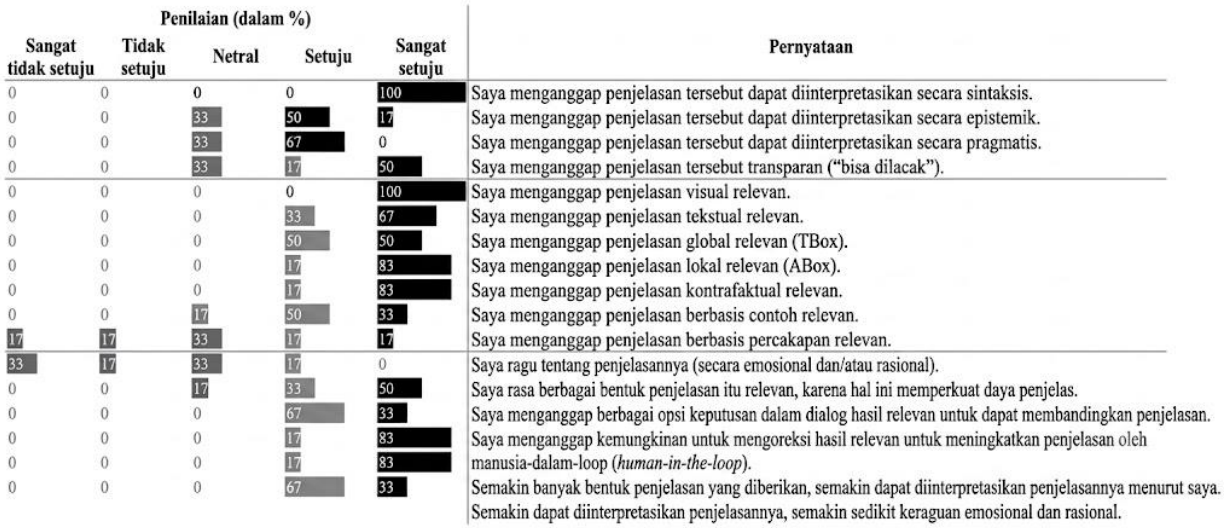
Probabilitas jalur antara MES (v_{i21}) serta ERP yang digunakan untuk mengurangi tugas manual (v_{i22}) mewakili hasil konkret dan menunjukkan jumlah jalur yang dilalui lebih tinggi dalam grafik pengetahuan. Dengan mempertimbangkan kelemahan transfer data dokumen analog (v_{21}) dan kerusakan sistem TI (v_{22}), keduanya tidak berbeda dalam contoh yang ditunjukkan. Alasannya adalah bahwa elemen solusi yang diusulkan mengejar tujuan yang sama dengan probabilitas bersyarat yang sama (ukuran peningkatan), lihat Gambar 6.9. Sistem TI ERP (v_{2211}) serta MES (v_{2212}) digunakan untuk mengurangi tugas manual (v_{i1}) dan untuk meningkatkan dukungan TI (vi2). Dalam kasus seperti itu, pemberian bobot melalui umpan balik ahli dapat memvalidasi kesesuaian solusi yang diberikan. Hasil yang lebih sesuai menerima bobot yang lebih tinggi agar dipertimbangkan lebih kuat dalam analisis lanjutan dengan jalur hasil yang serupa. Ini mewakili bentuk pembelajaran mesin interaktif, yang layak diselidiki dalam siklus DSR lainnya.

Pemilik proses dari proses yang dipertimbangkan mengevaluasi interpretasi yang dihasilkan (ditunjukkan pada Tabel 6.4) sebagai mudah dibaca secara sintaksis, dapat dipahami (epistemik) dan baginya masuk akal (pragmatis) karena proses yang dianalisis sudah familiar baginya. Mengingat probabilitas jalur yang lebih tinggi dari langkah-langkah peningkatan spesifik v_{21} dan v_{22}, ia setuju dengan interpretasi yang lebih baik dibandingkan dengan v_{i31}. Hal ini disebabkan oleh perbedaan antara hasil abstrak dan hasil spesifik proses.

Dengan demikian, langkah-langkah peningkatan spesifik proses mencakup lebih banyak interpretasi daripada yang abstrak, yang memiliki efek yang sama pada tingkat interpretasi. Dengan demikian, probabilitas jalur yang lebih rendah juga dikaitkan dengan interpretasi yang lebih rendah, seperti untuk elemen solusi abstrak dari lapisan abstraksi (L1), yang lebih umum dan oleh karena itu tidak mewakili alternatif solusi spesifik dibandingkan dengan v_{21} dan v_{22}. Namun, rasio antara node yang diaktifkan dan semua elemen penerus yang bergantung secara semantik sangat menentukan ruang lingkup interpretasi.

Siklus DSR terpisah, yang saat ini sedang berlangsung, berfokus pada investigasi dan pengembangan antarmuka penjelasan untuk sistem XAI [Antarmuka Pengguna yang Dapat Dijelaskan (XUI)]. Dalam konteks ini, enam peserta independen mengevaluasi prototipe pendekatan KBX-PA penulis menggunakan contoh studi kasus tunggal. Fokusnya adalah pada antarmuka pengguna, bentuk penjelasan, dan kegunaan serta penerimaannya oleh pengguna.

Pernyataan individu para ahli diringkas dan dievaluasi sebagai pernyataan kunci, seperti yang ditunjukkan pada Gambar 6.10.



Penilaian (dalam %)					Pernyataan
Sangat tidak setuju	Tidak setuju	Netral	Setuju	Sangat setuju	
0	0	0	0	100	Saya menganggap penjelasan tersebut dapat diinterpretasikan secara sintaksis.
0	0	33	50	17	Saya menganggap penjelasan tersebut dapat diinterpretasikan secara epistemik.
0	0	33	67	0	Saya menganggap penjelasan tersebut dapat diinterpretasikan secara pragmatis.
0	0	33	17	50	Saya menganggap penjelasan tersebut transparan ("bisa dilacak").
0	0	0	0	100	Saya menganggap penjelasan visual relevan.
0	0	0	33	67	Saya menganggap penjelasan tekstual relevan.
0	0	0	50	50	Saya menganggap penjelasan global relevan (TBox).
0	0	0	17	83	Saya menganggap penjelasan lokal relevan (ABox).
0	0	0	17	83	Saya menganggap penjelasan kontrafaktual relevan.
0	0	17	50	33	Saya menganggap penjelasan berbasis contoh relevan.
17	17	33	17	17	Saya menganggap penjelasan berbasis percakapan relevan.
33	17	33	17	0	Saya ragu tentang penjelasannya (secara emosional dan/atau rasional).
0	0	17	33	50	Saya rasa berbagai bentuk penjelasan itu relevan, karena hal ini memperkuat daya penjelas.
0	0	0	67	33	Saya menganggap berbagai opsi keputusan dalam dialog hasil relevan untuk dapat membandingkan penjelasan.
0	0	0	17	83	Saya menganggap kemungkinan untuk mengoreksi hasil relevan untuk meningkatkan penjelasan oleh manusia-dalam-loop (*human-in-the-loop*).
0	0	0	17	83	
0	0	0	67	33	Semakin banyak bentuk penjelasan yang diberikan, semakin dapat diinterpretasikan penjelasannya menurut saya.
					Semakin dapat diinterpretasikan penjelasannya, semakin sedikit keraguan emosional dan rasional.

Gambar 6.10 Peringkat pernyataan kunci yang dibuat dalam wawancara ahli.

Menjadi jelas bahwa bentuk-bentuk interpretasi menerima persetujuan yang meningkat seiring dengan menurunnya kompleksitas (pragmatis, epistemik, sintaksis). Dengan demikian, semua responden menganggap bahwa penjelasan tersebut dapat dipahami secara sintaksis, 67% peserta setuju dengan interpretasi epistemik (di mana 17% sangat setuju), dan juga 67% setuju dengan interpretasi pragmatis. Penilaian relevansi berbagai bentuk penjelasan menunjukkan bahwa penjelasan visual, kontrafaktual, dan lokal (ABox) menerima persetujuan terkuat, diikuti oleh penjelasan tekstual dan global (TBox), serta penjelasan berbasis contoh. Penjelasan berbasis obrolan dinilai tidak relevan maupun tidak tidak relevan.

Separuh peserta memiliki keraguan tentang penjelasan tersebut. Mayoritas berpendapat bahwa berbagai bentuk penjelasan dapat meningkatkan interpretasi penjelasan. Lebih dari 80% menganggap koreksi penjelasan oleh manusia sebagai hal yang relevan untuk meningkatkan penjelasan. Lebih lanjut, 83% setuju dengan pernyataan bahwa berbagai bentuk penjelasan berkontribusi pada interpretasi yang lebih tinggi. Tingkat persetujuan yang sedikit lebih rendah dicapai dengan pernyataan: semakin mudah diinterpretasikan penjelasannya, semakin sedikit keraguan yang ada terhadap penjelasan yang dihasilkan.

Evaluasi menunjukkan bahwa berbagai bentuk penjelasan dalam antarmuka penjelasan sistem XAI memungkinkan interpretasi hasil yang lebih tinggi. Selain penjelasan tekstual, penjelasan visual, serta penjelasan kontrafaktual tampaknya membantu meminimalkan keraguan pengguna terhadap penjelasan tersebut. Upaya penelitian baru dapat dikaitkan dengan hal ini, di mana interpretasi diukur menggunakan berbagai bentuk penjelasan dan dievaluasi sesuai kebutuhan.

6.6 INTEGRASI KNOWLEDGE GRAPH DAN MACHINE LEARNING UNTUK XAI-PA

Pendekatan KBX-PA dapat memajukan keadaan analisis proses otomatis dengan integrasi berbagai masukan untuk memastikan kelengkapan informasi, representasi pengetahuan, dan algoritma deduksi yang melakukan analisis proses semantik, dan akhirnya kesempatan untuk menginterpretasikan hasil kepada pengguna. Arsitektur grafik pengetahuan memungkinkan pembuatan model yang dapat diinterpretasikan dengan memetakan jalur hasil dalam bentuk pohon keputusan stokastik. Semantik yang dikodekan asli tidak hilang karena kelas asosiasi tetap dan algoritma deduksi terkait. Dengan cara ini, asal elemen solusi dapat dibuat mudah dibaca, dipahami, dan bahkan masuk akal bagi pengguna. Probabilitas jalur menunjukkan tingkat interpretasi solusi yang diusulkan dengan mempertimbangkan kelemahan yang diidentifikasi. Menggabungkan berbagai bentuk penjelasan, seperti visual, tekstual, kontrafaktual, atau berbasis contoh, dapat meningkatkan interpretasi penjelasan kepada pengguna, dan dengan demikian berkontribusi untuk meningkatkan daya penjelasan pendekatan XAI berbasis pengetahuan. Hal ini menciptakan nilai tambah yang signifikan dan, yang terpenting, relevan secara praktis untuk analisis proses otomatis.

Hasil yang dapat diinterpretasikan penting dalam praktik konsultasi, karena merupakan dasar penerimaan klien. Untuk konsultasi proses, KBX-PA menawarkan bentuk layanan digital. KBX-PA dapat digunakan secara mandiri sebagai layanan konsultasi mandiri oleh klien atau mempercepat analisis proses dan mengamankan kualitas hasil secara independen dari konsultan yang terlibat dalam proyek konsultasi klasik. Pada tingkat yang lebih umum, kepercayaan dan penerimaan terhadap sistem berbasis AI dapat meningkat secara signifikan ketika sistem tersebut mampu menjelaskan hasilnya kepada pengguna.

Penyempurnaan hasil yang berkelanjutan dan pembelajaran dapat dicapai melalui deduksi otomatis dan penerapan berulang. Sistem dapat ditingkatkan melalui pembelajaran mesin interaktif yang memperkaya grafik pengetahuan, menggunakan umpan balik dari konsultan proses yang berpengalaman. Hasil yang terverifikasi dipertimbangkan dan diprioritaskan oleh algoritma deduksi. Jika plausibilitas divalidasi secara eksternal melalui umpan balik, penerimaan hasil yang diinterpretasikan dan penggunaannya dalam proses pengambilan keputusan meningkat. Hal ini mengarah pada tujuan penelitian lebih lanjut, yang menyelidiki pengaruh peran manusia dalam proses interpretasi hasil.

Dalam konteks analisis proses, tampaknya perlu untuk menyelidiki sejauh mana knowledge graph cocok untuk meningkatkan interpretasi hasil analisis process mining. Model proses yang dibuat dengan metode process mining dapat berfungsi sebagai input untuk sistem kami. Dengan bantuan kerangka kerja XAI yang diuraikan untuk interpretasi, hasil analisis proses yang lebih mudah diinterpretasikan dapat memfasilitasi adopsi teknologi process mining dalam praktik.

Mengenai pendekatan XAI yang ada, menghasilkan model yang kaya dan mudah diinterpretasikan dari arsitektur knowledge graph dapat meningkatkan interpretasi hasil bagi pengguna. Pengembangan antarmuka penjelasan untuk sistem XAI membantu memberikan model yang dihasilkan yang memberikan penjelasan kepada pengguna dengan cara yang mudah dipahami manusia dan dapat mengurangi keraguan tentang penjelasan. Kombinasi

metode pembelajaran mesin berkinerja tinggi, seperti jaringan saraf dalam, dan pendekatan penjelasan berbasis knowledge graph penulis memiliki potensi besar untuk saling menguntungkan, sehingga mengatasi ketegangan antara kinerja dan interpretasi hasil AI.

BAB 7
PEMBELAJARAN MENDALAM

Pembelajaran mendalam adalah cabang dari pembelajaran mesin yang mencoba memodelkan abstraksi data tingkat tinggi menggunakan beberapa lapisan neuron yang terdiri dari struktur kompleks atau transformasi non-linier. Dengan meningkatnya jumlah data dan daya komputasi, jaringan saraf dengan struktur yang lebih kompleks telah menarik perhatian luas dan diterapkan di berbagai bidang. Pada bab ini ini memberikan gambaran umum tentang pembelajaran mendalam dalam jaringan saraf termasuk model arsitektur populer dan algoritma pelatihan.

7.1 PERKEMBANGAN AI: DARI MACHINE LEARNING HINGGA DEEP LEARNING

Gagasan untuk menciptakan mesin yang berpikir memiliki sejarah yang sangat panjang yang dapat ditelusuri kembali ke zaman Yunani kuno. Penemuan komputer yang dapat diprogram pada tahun 1940-an membuat para ilmuwan mulai serius mengeksplorasi kemungkinan membangun otak elektronik. Pada tahun 1956, sebuah pertemuan yang diadakan di Dartmouth College secara resmi menetapkan bidang penelitian Kecerdasan Buatan. Pada saat yang sama, jutaan dolar diinvestasikan dalam penelitian AI.

Akhirnya, para peneliti menyadari bahwa mereka sangat meremehkan kesulitan proyek ini. Dalam 20 tahun pertama penelitian AI, orang berpikir bahwa selama mesin memiliki kemampuan penalaran logis, mereka dapat menjadi cerdas. AI awal dapat memecahkan masalah matematika yang sangat kompleks yang dapat dijelaskan dengan daftar aturan matematika formal seperti bermain catur, tetapi tantangan sebenarnya adalah memecahkan masalah yang dapat dipecahkan manusia secara intuitif, seperti pengenalan gambar atau ucapan.

Untuk memecahkan masalah yang lebih intuitif ini, penalaran logis saja tidak cukup. Para ilmuwan mulai percaya bahwa kita harus mencoba membuat mesin mempelajari pengetahuan. Pembelajaran mesin adalah perancangan dan analisis algoritma yang memungkinkan komputer untuk secara otomatis mempelajari pengetahuan. Algoritma pembelajaran mesin secara otomatis menganalisis dan mengekstrak pola yang diperoleh dari data, dan menggunakan pola data untuk membuat prediksi.

Dari perspektif model hierarkis, perkembangan pembelajaran mesin telah melalui dua langkah utama sejak akhir tahun 1980-an: pembelajaran dangkal dan pembelajaran mendalam. Pada akhir tahun 1980-an, penemuan algoritma back propagation untuk jaringan saraf tiruan memicu gelombang pembelajaran mesin berdasarkan model statistik. Ditemukan bahwa penggunaan algoritma back propagation memungkinkan model jaringan saraf tiruan untuk mempelajari pola dari banyak data pelatihan, dan membuat prediksi berdasarkan pola-pola tersebut.

Metode ini mengalahkan sistem sebelumnya dalam banyak hal. Jaringan saraf tiruan pada saat itu adalah model dangkal yang hanya berisi satu lapisan tersembunyi. Pada tahun

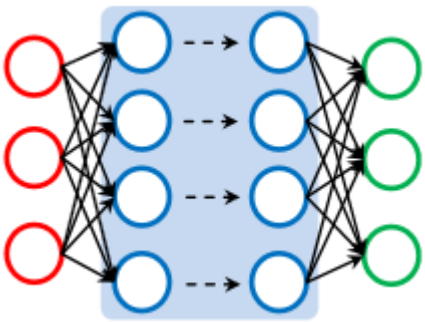
1990-an, berbagai model pembelajaran mesin dangkal telah diusulkan, seperti support vector machines (SVM), Boosting, metode entropi maksimum termasuk Regresi Logistik (LR) dan sebagainya. Model-model tersebut memiliki satu lapisan tersembunyi (seperti SVM, Boosting), atau tanpa lapisan tersembunyi (seperti LR). Model-model ini menunjukkan keberhasilan besar baik dalam analisis teoritis maupun aplikasi. Sebaliknya, karena kompleksitas analisis teoritis, dan persyaratan yang tinggi untuk pengalaman dan keterampilan, jaringan saraf tiruan relatif tidak populer pada periode ini.

Sejak tahun 2000, perkembangan pesat Internet telah menghadirkan permintaan besar untuk analisis dan prediksi cerdas dari sejumlah besar data. Pembelajaran mesin dangkal sangat sukses di bidang ini. Aplikasi yang paling sukses termasuk estimasi CTR untuk sistem pencarian iklan, sistem peringkat halaman, sistem penyaringan spam, sistem rekomendasi, dan sebagainya.

Pada tahun 2006, sebuah artikel oleh Geoffrey Hinton membuka gerbang pembelajaran mendalam di dunia akademis dan industri. Ada dua pesan utama dalam artikel ini: jaringan saraf tiruan dengan lebih banyak lapisan tersembunyi memiliki kemampuan belajar yang lebih baik; kesulitan dalam melatih jaringan saraf dalam dapat diatasi dengan pra-pelatihan per lapisan.

Sejak saat itu, penelitian tentang pembelajaran mendalam terus memanas. Sejak tahun 2011, Microsoft dan Google telah mengerjakan pengenalan suara menggunakan jaringan saraf dalam dan berhasil mengurangi tingkat kesalahan sebesar 20% hingga 30%, yang merupakan terobosan besar di bidang pengenalan suara. Pada tahun 2012, teknologi jaringan saraf dalam telah mencapai hasil yang luar biasa di bidang pengenalan gambar, dan hasil evaluasi di ImageNet menunjukkan bahwa tingkat kesalahan dapat dikurangi dari 26% menjadi 15%.

Saat ini, Google, Microsoft, dan perusahaan teknologi tinggi terkenal lainnya dengan data besar sangat ingin mengalokasikan sumber daya untuk studi pembelajaran mendalam, karena mereka percaya bahwa model yang lebih kompleks dan kuat dapat mengungkap kekayaan informasi dengan jumlah data yang sangat besar, dan membuat prediksi yang akurat untuk peristiwa masa depan atau yang tidak diketahui.



Gambar 7.1. Contoh sederhana jaringan saraf dalam (*deep neural network*).

Berbeda dengan pembelajaran mesin dangkal (*shallow machine learning*), pembelajaran mendalam (*deep learning*) menggunakan serangkaian lapisan unit pemrosesan nonlinier untuk ekstraksi dan transformasi fitur. Hal ini memungkinkan komputer untuk belajar dari representasi hierarkis data di mana fitur tingkat yang lebih tinggi diturunkan dari fitur tingkat yang lebih rendah. Gambar 7.1 memberikan contoh sederhana jaringan saraf dalam, di mana input algoritma pembelajaran mendalam ditransformasikan melalui beberapa lapisan tersembunyi dan output diturunkan dari komputasi lapisan tersembunyi.

Misalnya, jika diberikan sebuah gambar, jaringan saraf dalam dapat mengekstrak berbagai fitur dari gambar ini di setiap lapisan tersembunyi. Misalnya, jika diberikan piksel gambar ini sebagai input, tepi dapat diidentifikasi di lapisan pertama dengan membandingkan warna atau kecerahan piksel tetangga. Kemudian, lapisan tersembunyi kedua dapat mendeteksi sudut dan kontur menggunakan deskripsi tepi. Setelah itu, bagian-bagian dari objek tertentu dapat ditemukan dengan mencari kumpulan kontur dan sudut tertentu. Akhirnya, objek dalam gambar input dapat dikenali.

Pembelajaran mendalam telah menarik banyak perhatian dari kalangan akademisi hingga industri. Dalam aplikasi, pembelajaran mendalam telah membuat kemajuan besar dalam pengenalan suara dan gambar, sehingga mendorong kemajuan kecerdasan buatan dan interaksi manusia-komputer. Di sisi lain, masih ada beberapa pertanyaan tentang pembelajaran mendalam yang perlu kita pikirkan.

Dibandingkan dengan model dangkal, model mendalam memiliki kemampuan representasi fungsi nonlinier yang lebih baik. Namun, sulit untuk menghitung berapa banyak sumber daya yang kita butuhkan untuk mendapatkan model mendalam yang lebih baik melalui pelatihan dan untuk menemukan algoritma optimasi yang ideal. Karena model mendalam bersifat non-konveks, penelitian teoritis di bidang ini mungkin sulit. Di industri, bagaimana memanfaatkan komputasi paralel masif untuk melakukan pelatihan data masif adalah masalah yang sangat penting. Teknik pelatihan jaringan saraf dalam yang ada sebagian besar menggunakan metode gradien stokastik (SGD). Metode ini sayangnya tidak dapat dijalankan secara paralel.

7.2 ARSITEKTUR PEMBELAJARAN MENDALAM

Penelitian pembelajaran mendalam berupaya memodelkan data skala besar dengan beberapa lapisan pemrosesan dengan struktur yang kompleks. Dengan demikian, tidak seperti arsitektur pembelajaran mesin dangkal, arsitektur pembelajaran mendalam terdiri dari beberapa transformasi non-linier seperti yang ditunjukkan pada Gambar 7.1.

Terdapat sejumlah besar varian arsitektur mendalam, dan arsitektur yang berbeda dapat digunakan untuk merepresentasikan sumber data yang berbeda. Misalnya, jaringan saraf konvolusional adalah arsitektur yang paling populer untuk pengenalan gambar, dan jaringan saraf berulang lebih cocok untuk tugas-tugas sekuensial seperti pengenalan tulisan tangan atau ucapan. Pada bagian ini, kita akan memperkenalkan arsitektur-arsitektur ini dan membahas mengapa arsitektur tersebut cocok untuk menangani data yang berbeda.

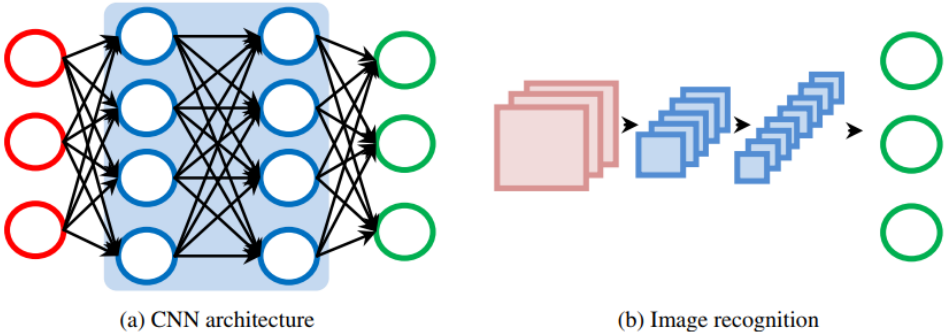
Jaringan saraf konvolusional (CNN)

Jaringan saraf konvolusional (CNN) adalah jenis jaringan saraf tiruan feed-forward yang menggunakan konvolusi di setidaknya salah satu lapisannya. CNN terinspirasi oleh jaringan saraf biologis. CNN menggabungkan jaringan saraf tiruan dan konvolusi diskrit untuk pemrosesan gambar yang dapat digunakan untuk mengekstrak fitur secara otomatis. Dengan demikian, CNN dirancang khusus untuk mengenali data dua dimensi, seperti gambar dan video. Gambar dapat langsung digunakan sebagai input jaringan, yang menghindari proses ekstraksi fitur dan rekonstruksi data yang kompleks dalam algoritma pengenalan gambar tradisional.

Pada tahun 1980, Kunihiko Fukushima mengusulkan model jaringan saraf untuk mekanisme pengenalan pola visual — "neokognitron", yang merupakan pendahulu CNN. Setelah itu, para peneliti mencoba berbagai cara untuk melatih jaringan saraf multi-lapisan, dan menerapkannya pada berbagai skenario. Namun, sementara lebar dan kedalaman jaringan meningkat, kinerja algoritma CNN dibatasi oleh sumber daya komputasi. Setelah tahun 2006, GPU yang efisien menjadi perangkat komputasi umum, yang memungkinkan pelatihan jaringan yang lebih besar. Dengan demikian, beberapa model telah diusulkan untuk melatih jaringan saraf konvolusional dalam secara lebih efisien. Saat ini, CNN digunakan di banyak bidang termasuk pemahaman gambar dan video.

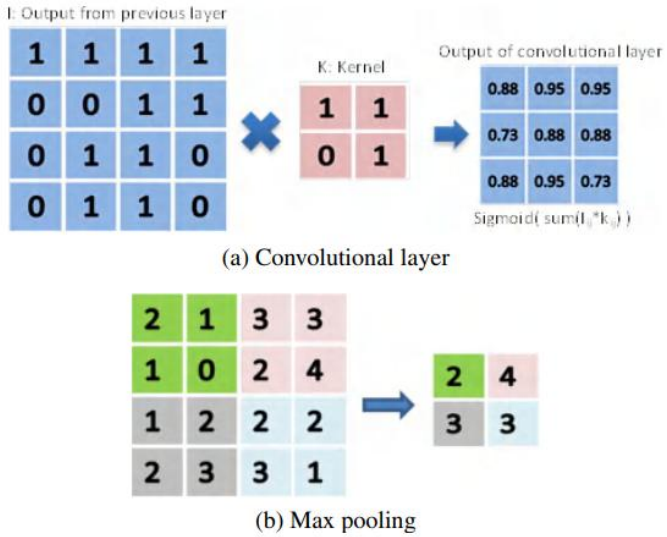
aringan Saraf Konvolusional dimodelkan sebagai kumpulan unit neuron yang terhubung dalam grafik asiklik. Gambar 7.2(a) menunjukkan contoh jaringan saraf konvolusional dengan 2 lapisan tersembunyi. Setiap lapisan jaringan ini mengambil output dari lapisan sebelumnya sebagai input. Oleh karena itu, siklus tidak diperbolehkan dalam Arsitektur CNN. Biasanya, unit neuron antara dua lapisan yang berdekatan terhubung sepenuhnya, tetapi neuron dalam satu lapisan tidak berbagi koneksi. Jaringan saraf dalam (*deep neural network*) berisi lebih dari 2 lapisan.

Mungkin ada 20 lapisan atau lebih dan setiap lapisan dapat berbeda secara fungsional. Ambil contoh pengenalan gambar, di mana CNN mengatur neuronnya dalam tiga dimensi (lebar, tinggi, kedalaman), seperti yang ditunjukkan pada Gambar 7.2(b). Input CNN pada Gambar 7.2(b) adalah gambar, sehingga lebar dan tingginya adalah dimensi gambar, dan kedalamannya adalah 3 (saluran Merah, Hijau, Biru).



Gambar 7.2. Jaringan saraf konvolusional.

Output CNN adalah vektor tunggal skor kelas, yang disusun sepanjang dimensi kedalaman. Antara input dan output, tiga jenis lapisan utama digunakan untuk membangun arsitektur CNN termasuk lapisan konvolusi, lapisan pooling, dan lapisan yang terhubung sepenuhnya. Dan ada juga fungsi kerugian (misalnya SVM/Softmax) di lapisan terakhir. Terkadang, jenis lapisan lain digunakan untuk meningkatkan kinerja CNN seperti lapisan dropout yang digunakan untuk mengontrol ukuran CNN.



Gambar 7.3. Lapisan-lapisan berbeda dari jaringan saraf konvolusional.

Untuk meringkas secara lebih rinci:

- *Lapisan input* menyimpan nilai piksel mentah dari gambar.
- *Lapisan konvolusi* meningkatkan fitur data input dan mengurangi noise. Input dari lapisan konvolusi adalah output dari lapisan sebelumnya. Untuk setiap input, terdapat kernel dengan bobot di dalam sel-selnya. Kernel di dalam lapisan konvolusi memiliki ukuran yang sama. Seperti yang ditunjukkan pada Gambar 7.3(a), ukuran kernel adalah $2 * 2$, dan input dari lapisan konvolusi ini adalah $4 * 4$. Kita menggerakkan kernel dari kiri atas ke kanan bawah input dengan langkah = 1 dan mendapatkan output dengan ukuran $(4 - 2 + 1) * (4 - 2 + 1) = 9$. Setiap angka dalam output ini dihitung dengan dua langkah. Langkah pertama adalah menghitung penjumlahan angka berbobot yang dicakup oleh kernel. Langkah kedua adalah fungsi sigmoid yang memampatkan hasil langkah pertama menjadi pecahan yang berkisar antara 0 dan 1. Ukuran keluaran lapisan konvolusi ditentukan oleh ukuran masukan dan kernel. Jumlah keluaran telah ditentukan sebelumnya oleh pengguna dan menentukan jumlah kernel.
- *Lapisan pooling* mengambil subsampel dari data masukan. Lapisan pooling membagi masukan menjadi beberapa wilayah kecil, dan melakukan beberapa fungsi seperti rata-rata atau maksimum pada setiap wilayah. Lapisan max pooling ditunjukkan pada Gambar 7.3(b). Masukan dengan ukuran $4 * 4$ dibagi menjadi empat wilayah dengan ukuran $2 * 2$, dan keluarannya adalah angka maksimum dari keempat wilayah ini. Tidak seperti lapisan konvolusi, tidak ada tumpang tindih dalam perhitungan lapisan pooling.

- *Lapisan fully-connected* mengambil semua neuron dari lapisan sebelumnya dan menghubungkannya ke setiap neuron yang dimilikinya. Lapisan yang terhubung sepenuhnya tidak lagi berlokasi secara spasial (kita dapat memvisualisasikannya sebagai satu dimensi), sehingga tidak mungkin ada lapisan konvolusi setelah lapisan yang terhubung sepenuhnya.

Untuk mengekstrak pola yang tepat dari data, kita perlu melatih jaringan. Proses pelatihan adalah memperbarui bobot dalam jaringan untuk meminimalkan kesalahan yang dihasilkan dari jaringan. Kita akan memperkenalkan beberapa algoritma pelatihan di Bagian selanjutnya.

Jaringan saraf berulang (RNN)

RNN bertujuan untuk memproses data sekuensial. Dalam model jaringan saraf feed forward tradisional, seperti CNN, data mengalir dari lapisan input ke lapisan tersembunyi dan kemudian ke lapisan output. Tidak ada koneksi antar neuron di lapisan yang sama. Beberapa masalah tidak dapat sepenuhnya ditangani oleh jenis jaringan saraf ini. Arsitektur ini tidak dapat menyelesaikan masalah di mana data input memiliki hubungan satu sama lain.

Misalnya, kita membutuhkan kata sebelumnya dalam sebuah kalimat untuk memprediksi kata berikutnya, karena kata-kata dalam kalimat yang sama tidak independen. Dalam RNN, output saat ini dan output sebelumnya relevan. Lebih spesifiknya, output dari langkah sebelumnya disimpan dan digunakan untuk menghitung output saat ini, yaitu, input jaringan berisi data dari lapisan input dan output dari lapisan tersembunyi dari langkah sebelumnya. Hal ini membuat mereka dapat diterapkan pada tugas-tugas seperti pengenalan tulisan tangan dan pengenalan ucapan. Arsitektur RNN ditunjukkan pada Gambar 7.4.

Pada Gambar 7.4, selain aliran data dari input ke lapisan tersembunyi dan dari lapisan tersembunyi ke output, yang sama seperti pada CNN, terdapat juga transfer data antar neuron di lapisan tersembunyi yang sama. Jaringan saraf berulang dapat diuraikan menjadi jaringan saraf asiklik, seperti yang ditunjukkan pada Gambar 7.5. Misalnya, jaringan saraf berulang dari kalimat lima kata dapat diuraikan menjadi jaringan saraf lima lapis dengan setiap lapisan mewakili satu kata.

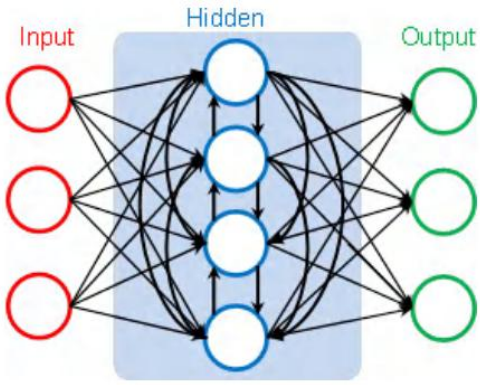
Pada Gambar 7.5, input dan output dari jaringan saraf berulang dilambangkan sebagai x dan y. t mewakili jumlah langkah. Misalnya, x_1 dan y_1 adalah input dan output dari langkah 1, di mana x_1 mungkin merupakan kata pertama dalam satu kalimat, dan y_1 adalah kata berikutnya yang diprediksi. s_t, yang merupakan keadaan jaringan pada langkah t, dapat dihitung menggunakan input dari langkah t x_t dan output dari lapisan tersembunyi pada langkah sebelumnya s_{t-1} dengan

$$s_t = f(U * x_t + W * s_{t-1}) \tag{7.1}$$

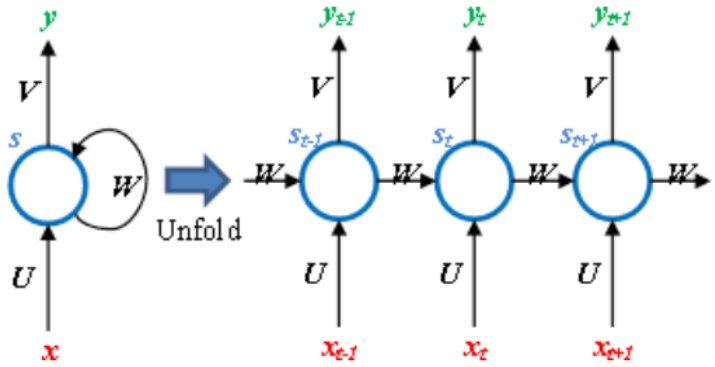
di mana f biasanya merupakan fungsi aktivasi nonlinier. s_t juga dapat dilihat sebagai unit memori di lapisan tersembunyi yang dapat digunakan untuk menyimpan informasi dari langkah-langkah sebelumnya.

Dalam jaringan saraf konvolusional, parameter dari lapisan yang berbeda berbeda. Namun, dalam RNN, setiap lapisan berbagi parameter U, V, dan W. Itu berarti setiap langkah

dalam RNN melakukan hal yang sama, hanya dengan input yang berbeda. Ini sangat mengurangi kebutuhan untuk mempelajari parameter jaringan.



Gambar 7.4. Jaringan saraf berulang.



Gambar 7.5. Jaringan saraf berulang yang terbentang.

Selama bertahun-tahun, para peneliti telah mengusulkan banyak RNN canggih untuk mengatasi kekurangan model RNN biasa. Berikut adalah beberapa model RNN yang banyak digunakan:

- *RNN Sederhana* adalah kasus khusus dari RNN, yang merupakan jaringan tiga lapis, dan menambahkan unit konteks di dalam lapisan tersembunyi. Koneksi antara unit konteks dan neuron tetap, begitu pula bobotnya. Pada setiap langkah, ia dilatih seperti yang kita lakukan dengan jaringan saraf feed forward. Output dari neuron disimpan ke unit konteks yang terhubung dengannya, dan kemudian digunakan sebagai input neuron untuk langkah selanjutnya.
- *Deep Bidirectional RNN* didasarkan pada gagasan bahwa output saat ini tidak hanya terkait dengan elemen sebelumnya, tetapi juga terkait dengan elemen di masa mendatang. Misalnya, dalam kasus prediksi kata, kita akan membutuhkan konteks kiri dan kanan untuk memprediksi kata yang hilang dalam satu kalimat. Bidirectional RNN adalah RNN yang relatif sederhana, yang terdiri dari dua RNN yang ditumpuk di atas satu sama lain, dan output ditentukan oleh status lapisan tersembunyi di kedua RNN ini.

- *Echo State Network* adalah jenis RNN khusus. Gagasan dasarnya adalah menggunakan jaringan siklus penghubung acak skala besar untuk menggantikan lapisan tersembunyi dalam jaringan saraf klasik, yang menyederhanakan proses pelatihan.
- *Gated Recurrent Unit Recurrent Neural Networks* memberikan bobot yang berbeda pada keadaan sebelumnya sesuai dengan perbedaannya dengan keadaan saat ini, dan hanya beberapa bobot yang diperbarui untuk meminimalkan fungsi kerugian.
- *Long short-term memory* (LSTM) sangat populer akhir-akhir ini. RNN dapat menangani masalah ketika ada beberapa hubungan atau ketergantungan di antara data masukan. Namun, RNN mungkin kehilangan kemampuan untuk mempelajari koneksi di antara data masukan ketika ketergantungannya sangat panjang. Misalnya, diberikan kalimat "awan berada di", RNN dapat memprediksi kata berikutnya adalah "langit", tetapi sulit baginya untuk memprediksi kata berikutnya dari kalimat "Saya tumbuh di Prancis. Saya berbicara fasih". LSTM diusulkan untuk menyelesaikan masalah ini. Dalam LSTM, neuron digantikan oleh blok. Di setiap blok, ada sel yang digunakan untuk mengingat informasi dari semua data masukan. Dan satu komponen bernama "gerbang lupa" digunakan untuk menghapus informasi yang tidak penting dari sebuah sel seperti "the" atau "of".
- *Clockwork RNNs* (CW-RNNs) membagi lapisan tersembunyi menjadi beberapa kelompok. Setiap kelompok memproses inputnya sesuai dengan frekuensi jamnya sendiri.

RNN telah terbukti sangat sukses dalam pemrosesan bahasa alami (NLP), termasuk pemodelan bahasa dan pembuatan teks, terjemahan mesin, dan pengenalan ucapan. Model RNN yang paling sukses dan banyak digunakan adalah LSTM.

Terdapat beberapa perbedaan antara RNN dan CNN. Pertama, parameter W, U, V digunakan bersama untuk RNN yang tidak terlipat. Kedua, output setiap langkah bergantung tidak hanya pada input saat ini, tetapi juga sejumlah langkah sebelumnya dari jaringan. Jadi orang menggunakan algoritma lain untuk melatih RNN yang disebut Backpropagation Through Time (BPTT).

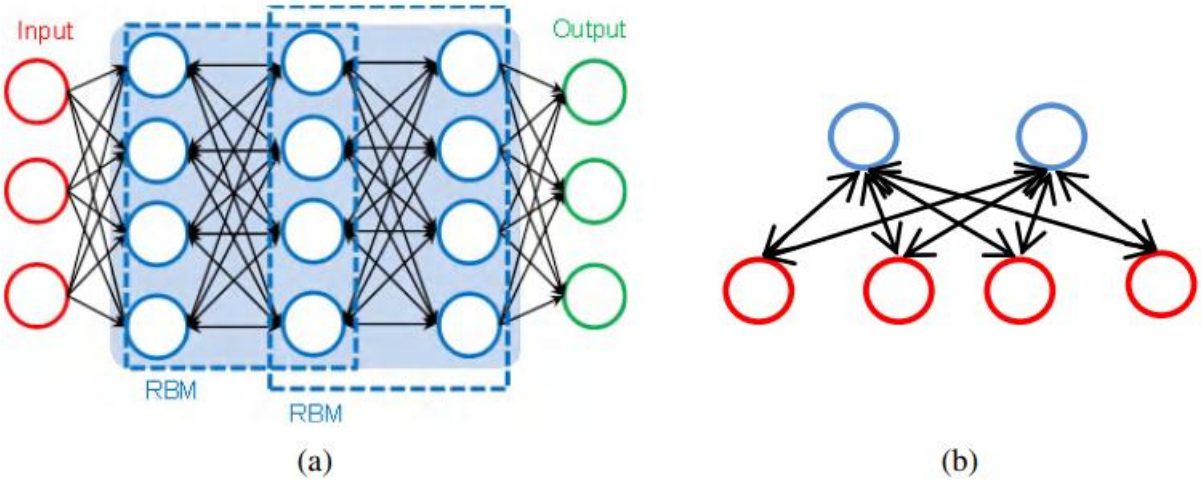
Jaringan Kepercayaan Mendalam (Deep Belief Networks/DBN)

Ide Jaringan Kepercayaan Mendalam (*Deep Belief Networks*/DBN) diusulkan oleh Geoffrey Hinton pada tahun 2006. DBN tidak hanya dapat digunakan untuk mengenali dan mengklasifikasikan data, tetapi juga dapat menghasilkan data. DBN terdiri dari beberapa lapisan neuron. Neuron-neuron ini dibagi menjadi unit tersembunyi dan unit terlihat. Unit terlihat digunakan untuk menerima data, sedangkan unit tersembunyi digunakan untuk mengekstrak fitur, sehingga disebut juga detektor fitur. Sebuah DBN terdiri dari beberapa lapisan *Restricted Boltzmann Machines* (RBM). Pelatihan DBN diproses lapis demi lapis. Di setiap lapisan, data masukan digunakan untuk menghitung unit tersembunyi, dan kemudian unit tersembunyi diberikan ke lapisan berikutnya sebagai data masukan. Gambar 7.6 menunjukkan contoh jaringan kepercayaan mendalam dan Restricted Boltzmann Machine.
Proses pelatihan DBN meliputi dua langkah utama:

Langkah 1: *Pra-pelatihan:* melatih RBM secara terpisah dengan cara tanpa pengawasan. Pastikan bahwa ketika vektor fitur dipetakan ke ruang yang berbeda, informasi dipertahankan sebanyak mungkin.

Langkah 2: Penyesuaian Halus: Menggunakan kesalahan dari lapisan keluaran untuk memperbarui bobot di DBN. Proses ini adalah pelatihan terawasi yang mirip dengan pelatihan CNN.

Dibandingkan dengan jaringan saraf tradisional, DBN dapat dilatih lapis demi lapis. Ini secara signifikan mengurangi sumber daya yang dibutuhkan untuk melatih jaringan.



Gambar 7.6. (a) Deep Belief Network dan (b) Restricted Boltzmann Machine.

Arsitektur Lainnya

Selain CNN, RNN, dan DBN, terdapat banyak arsitektur jaringan dalam lainnya termasuk:

- Deep stacking network (DSN) yang terdiri dari beberapa blok bertumpuk, di mana setiap modul blok sederhana dan mudah dilatih secara terawasi. DSN terutama digunakan untuk klasifikasi dan regresi gambar dan bahasa.
- *Deep coding network* (DPCN) yang merupakan skema pengkodean prediktif yang memprediksi representasi suatu lapisan dengan menggunakan pendekatan top-down.

Arsitektur lainnya termasuk Spike-and-slab RBM dan Deep Q-network.

7.3 ALGORITMA PELATIHAN

Seperti yang disebutkan di subBab, kita perlu melatih atau mengajarkan model pembelajaran mendalam sebelum kita dapat menggunakannya untuk melakukan klasifikasi atau prediksi. Pelatihan model pembelajaran mesin dapat diringkas menjadi beberapa kategori:

- *Pembelajaran terawasi* mempelajari suatu fungsi dari data pelatihan yang diberikan. Ketika data baru tiba, output dapat diprediksi sesuai dengan fungsi ini. Kumpulan data pelatihan pembelajaran terawasi mencakup input dan output yang dibutuhkan, yang biasanya diberi label oleh manusia.

- *Pembelajaran tak terawasi* belajar dari kumpulan data pelatihan yang tidak berlabel.
- *Pembelajaran semi-terawasi* belajar dari sejumlah kecil data berlabel dengan sejumlah besar data tak berlabel.
- *Pembelajaran penguatan* mempelajari cara melakukan tindakan melalui pengamatan. Setiap tindakan memiliki dampak pada lingkungan. Pembelajaran penguatan membuat penilaian berdasarkan umpan balik dari lingkungan yang diamati.

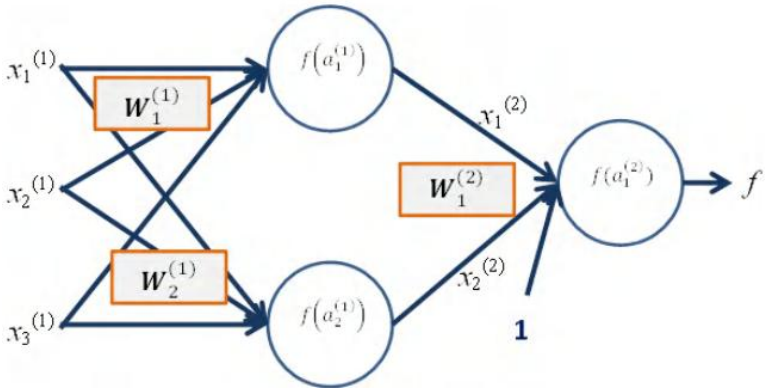
Pada bagian ini, kita akan membahas dua kategori pertama yang banyak digunakan dalam pelatihan model deep learning.

Pembelajaran Terawasi

Pembelajaran terawasi adalah melatih jaringan menggunakan sampel pelatihan yang ada termasuk data dan output yang sesuai. Pembelajaran terawasi biasanya digunakan untuk melakukan klasifikasi dan prediksi data.

Backpropagation (BP)

Algoritma backpropagation digunakan dalam banyak skenario pembelajaran mesin untuk melatih model pembelajaran. Algoritma ini awalnya diperkenalkan pada tahun 1970-an, tetapi pentingnya algoritma ini baru sepenuhnya disadari setelah karya David Rumelhart pada tahun 1986. Mereka menjelaskan beberapa jaringan saraf di mana backpropagation bekerja jauh lebih cepat daripada pendekatan sebelumnya. Saat ini, algoritma backpropagation adalah algoritma yang paling banyak digunakan untuk pembelajaran tak terawasi dalam jaringan saraf.



Gambar 7.7. Contoh jaringan saraf.

Pada bagian ini, kita akan memperkenalkan algoritma backpropagation menggunakan contoh kecil seperti yang ditunjukkan pada Gambar 7.7, yaitu jaringan saraf dengan dua lapisan tersembunyi. Dalam jaringan ini, setiap lingkaran menunjukkan neuron yang mencakup dua langkah perhitungan. Langkah pertama adalah menghitung penjumlahan berbobot dari data input. Kita menyebut hasil langkah ini sebagai aktivasi, dan menunjuknya sebagai a. Langkah kedua adalah fungsi sigmoid yang membatasi nilai a agar berada di antara 0 dan 1. Kita menggunakan f untuk mewakili fungsi ini.
Input dari lapisan pertama adalah vektor dengan 3 angka $X^{(1)} = \{x_1{}^{(1)}, x_2{}^{(1)}, x_3{}^{(1)}\}$, dan output dari lapisan ini adalah $f_1{}^{(1)}$ dan $f_2{}^{(1)}$, di mana:

$$f_1^{(1)} = f\left(a_1{}^{(1)}\right) = 1/(1 + e^{\left(-a_1{}^{(1)}\right)}) = 1/(1 + e^{\left(-(x_1{}^{(1)} * w_{11}{}^{(1)} + x_2{}^{(1)} * w_{12}{}^{(1)} + x_3{}^{(1)} * w_{13}{}^{(1)})\right)}) \quad (7.2)$$

$$f_2^{(1)} = f\left(a_2{}^{(1)}\right) = 1/(1 + e^{\left(-a_2{}^{(1)}\right)}) = 1/(1 + e^{\left(-(x_1{}^{(1)} * w_{21}{}^{(1)} + x_2{}^{(1)} * w_{22}{}^{(1)} + x_3{}^{(1)} * w_{23}{}^{(1)})\right)}) \quad (7.3)$$

dan output dari lapisan pertama menjadi input dari lapisan kedua, yaitu $x_1{}^{(2)} = f_1{}^{(1)}$ dan $x_2{}^{(2)} = f_2{}^{(1)}$ Angka input ketiga dari lapisan kedua adalah konstanta 1, yang disebut bias. Jadi output dari lapisan kedua adalah:

$$f_1{}^{(2)} = f\left(a_2{}^{(1)}\right) = 1/(1 + e^{\left(-a_1{}^{(2)}\right)}) = 1/(1 + e^{\left(-(x_1{}^{(2)} * w_{11}{}^{(2)} + x_2{}^{(2)} * w_{12}{}^{(2)} + w_{13}{}^{(2)})\right)}) \quad (7.4)$$

Misalkan kita memiliki empat item data dalam set pelatihan kita, seperti yang ditunjukkan pada Tabel 7.1 di mana $\{x_1, x_2, x_3\}$ adalah data masukan dan d adalah keluaran yang sesuai.

Pada lapisan keluaran, kita menggunakan fungsi kesalahan kuadrat: $\varepsilon = (d - f)^2$ untuk menunjukkan kesalahan jaringan saraf dan memperbarui bobot. Menurut Persamaan (7.2), (7.3), dan (7.4), diberikan masukan $X^{(1)} = \{x_1{}^{(1)}, x_2{}^{(1)}, x_3{}^{(1)}\}$, ε adalah fungsi dari semua bobot $w_{mi}{}^{(l)}$. Dengan demikian kita dapat memperoleh nilai minimum ε menggunakan turunan parsial ε terhadap $w_{mi}{}^{(l)}$: $\nabla\varepsilon\left(w_{mi}{}^{(l)}\right)$. Jika kita memperlakukan semua bobot kecuali $w_{m'i'}{}^{(l')}$ sebagai konstanta, $\varepsilon\left(w_{m'i'}{}^{(l')}\right)$ adalah fungsi satu dimensi, dan minimum dari fungsi ini adalah titik di mana $\nabla\varepsilon\left(w_{m'i'}{}^{(l')}\right) = 0$. Kita dapat menggunakan $\nabla\varepsilon\left(w_{m'i'}{}^{(l')}\right)$ untuk menentukan apakah kita perlu menambah atau mengurangi $w_{m'i'}{}^{(l')}$ untuk mendapatkan nilai minimum ε: jika $\nabla\varepsilon\left(w_{m'i'}{}^{(l')}\right)$ kurang dari 0, kita perlu menambah $w_{m'i'}{}^{(l')}$ dan jika $\nabla\varepsilon\left(w_{m'i'}{}^{(l')}\right)$ lebih besar dari 0, kita perlu mengurangi $w_{m'i'}{}^{(l')}$.
Jadi untuk setiap bobot kita dapat menyesuaikan $w_{mi}{}^{(l)}$ menggunakan:

$$w_{mi}{}^{(l)} - c_i{}^{(l)}\nabla\varepsilon(w_{mi}{}^{(l)}) \rightarrow w_{mi}{}^{(l)} \quad (7.5)$$

di mana $c_i{}^{(l)}$ adalah panjang langkah penyesuaian, yang menentukan berapa banyak langkah yang kita butuhkan sebelum mencapai minimum fungsi. Jika $c_i{}^{(l)}$ terlalu kecil, kita akan mengambil lebih banyak langkah untuk mendapatkan titik minimum fungsi. Jika $c_i{}^{(l)}$ terlalu besar, kita mungkin berayun antara kiri dan kanan titik minimum berkali-kali dan juga membutuhkan lebih banyak langkah untuk mendapatkan titik tersebut. Metode ini disebut proses penurunan gradien yang telah terbukti menjadi

salah satu cara paling efisien untuk mendapatkan kesalahan minimum untuk pembelajaran mesin.

Tabel 7.1. Data pelatihan.

x_1	x_2	x_3	d
1	0	1	0
0	0	1	1
0	1	1	0
1	1	1	1

Dalam contoh kita, kita akan memperbarui bobot dari lapisan terakhir ke lapisan pertama; inilah mengapa algoritma ini disebut backpropagation. Turunan lapisan terakhir dapat dihitung sebagai:

$$\nabla\varepsilon\left({w_{1i}}^{(2)}\right) = \partial\varepsilon/\delta {w_{1i}}^{(2)} = \delta(d-f)^2/\delta {w_{1i}}^{(2)} = -2(d-f)f(1-f) * {x_i}^{(2)} \qquad (7.6)$$

Jadi, dengan ukuran langkah ${c_i}^{(l)}$, kita dapat memperbarui bobot di lapisan kedua menggunakan Persamaan (7.6) dan keluaran lapisan pertama.
Derivasi lapisan pertama dapat dihitung sebagai:

$$\begin{aligned}\nabla\varepsilon\left({w_{mi}}^{(1)}\right) &= \delta(d-f)^2/\delta {w_{mi}}^{(1)} \\ &= -2(d-f) * \left(\delta f\left({a_m}^{(2)}\right)/\delta {a_m}^{(1)}\right) * (\delta {a_m}^{(1)}/\delta {w_{mi}}^{(1)})\end{aligned} \qquad (7.7)$$

di mana ${a_m}^{(1)} = {x_1}^{(1)} * {w_{m1}}^{(1)} + {x_2}^{(1)} * {w_{m2}}^{(1)} + {x_3}^{(1)} * {w_{m3}}^{(1)}$ menurut persamaan (2) dan (3). Dengan demikian

$$\delta {a_m}^{(1)}/\delta {w_{mi}}^{(1)} = {x_i}^{(1)} \qquad (7.8)$$

Menurut persamaan (7.2), (7.3) dan (7.4):

$$\begin{aligned}\delta f\left({a_m}^{(2)}\right)/\delta {a_m}^{(1)} &= \left(\delta f\left({a_m}^{(2)}\right)/\delta {a_m}^{(1)}\right) * (\delta {a_m}^{(2)}/\delta {w_{mi}}^{(1)}) \\ &= f(1-f) * {w_{mi}}^{(2)} * (\delta {a_m}^{(1)}/\delta {a_m}^{(1)}) \\ &= f(1-f) * {w_{mi}}^{(2)} * {f_m}^{(1)} * \left(1-{f_m}^{(1)}\right)\end{aligned} \qquad (7.9)$$

Jadi, dengan mengetahui output dari lapisan pertama dan lapisan kedua, bobot pada lapisan pertama dapat diperbarui menggunakan bobot yang telah diperbarui pada lapisan kedua menggunakan Persamaan (7.7), (7.8), (7.9).

Biasanya, kita menggunakan semua data pelatihan untuk memperbarui bobot berkali-kali hingga kesalahan pada lapisan output dari semua data pelatihan kurang dari ambang batas tertentu. Tabel 7.2 merangkum algoritma backpropagation.

Tabel 7.2. Algoritma Backpropagation.

Backpropagation
Bobot acak awal Lakukan hingga kriteria penghentian terpenuhi: Untuk setiap input (x, y): Lakukan proses feed-forward untuk menghitung aktivasi di semua lapisan tersembunyi, lalu pada lapisan keluaran memperoleh keluaran y' Ukurlah penyimpangan y' dari target y. Sebarkan kesalahan secara balik melalui jaringan dan lakukan pembaruan bobot.

Backpropagation through time (BPTT)

BPTT adalah algoritma pembelajaran terawasi untuk melatih RNN. Algoritma ini mirip dengan algoritma backpropagation, kecuali bahwa bobot U, V, dan W dalam RNN dibagi oleh setiap langkah seperti yang ditunjukkan pada Gambar 7.5. Jadi kita perlu menjumlahkan kontribusi setiap langkah waktu terhadap gradien.

Keadaan dan output yang diperkirakan dari setiap langkah dalam RNN dilambangkan sebagai:

$$s_t = \tanh(U * x_t + W * s_{t-1})$$
$$y_t = \text{softmax}(V * s_t)$$

Dan fungsi kerugian didefinisikan sebagai kerugian entropi silang:

$$E(d_t, y_t) = -\Sigma d_t \log y_t$$

di mana d_t adalah kata yang benar pada langkah t, dan y_t adalah estimasi atau prediksi. Kita biasanya menggunakan total kesalahan dari semua kata dalam satu kalimat untuk memperbarui bobot. Sama seperti bagaimana kita menjumlahkan kesalahan, kita menjumlahkan gradien di setiap langkah waktu untuk satu sampel pelatihan:

$$\delta E/\delta W = \Sigma \delta E_t/\delta W$$

Algoritma BPTT ditunjukkan pada Tabel 7.3.

Tabel 7.3. Algoritma backpropagation through time.

Perambatan balik melalui waktu
Bobot acak awal Buka jaringannya Lakukan hingga kriteria penghentian terpenuhi: $\boldsymbol{x}$ = vektor dengan magnitudo nol untuk $\boldsymbol{t}$ dari $\mathbf{0}$ ton $\mathbf{1}$: Atur input jaringan ke $\boldsymbol{x}, \boldsymbol{a}[\boldsymbol{t}], \boldsymbol{a}[\boldsymbol{t}+\mathbf{1}], \ldots$

Lakukan proses feed-forward untuk menghitung aktivasi di semua lapisan tersembunyi, lalu pada lapisan keluaran memperoleh keluaran $\boldsymbol{y}'$ Ukurlah penyimpangan $\boldsymbol{y}'$ dari target $\boldsymbol{y}$. Lakukan backpropagasi kesalahan melalui jaringan dan perbarui bobot. Rata-ratakan bobot di setiap langkah, sehingga parameternya identik $\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})$

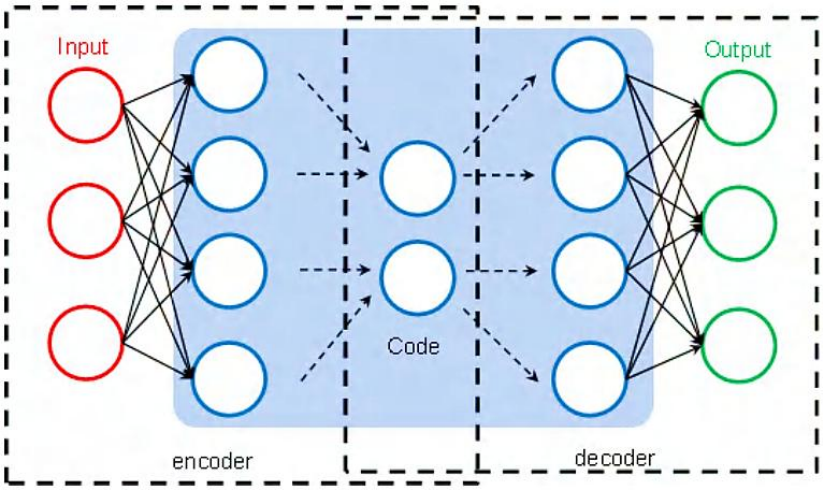
Pembelajaran Tanpa Pengawasan

Dalam pembelajaran tanpa pengawasan, kita memberikan algoritma kita sejumlah besar data tanpa label untuk mempelajari representasi fitur yang baik dari input. Pembelajaran tanpa pengawasan sering digunakan untuk melakukan pengelompokan dan pengkodean data.

Autoencoder

Autoencoder adalah salah satu algoritma pembelajaran tanpa pengawasan yang paling banyak digunakan untuk jaringan saraf dalam. Autoencoder hanyalah encoder kompresi, yang berfungsi untuk mengubah sesuatu dengan mengubah input menjadi output yang sama dengan input. Yaitu untuk menemukan fungsi yang memungkinkan fungsi $(\text{input}) = \text{input}$, yang disebut fungsi identitas. Gambar 7.8 menunjukkan struktur skematis dari sebuah autoencoder.

Intuisi di balik algoritma ini adalah bahwa jaringan tidak mempelajari data pelatihan antara tag dan "peta", tetapi sebaliknya mempelajari struktur data internal dan karakteristiknya sendiri (dan oleh karena itu, lapisan tersembunyi menjadi detektor fitur). Biasanya jumlah unit tersembunyi lebih sedikit daripada jumlah lapisan input/output, memaksa jaringan untuk hanya mempelajari fitur-fitur terpenting dan mencapai pengurangan dimensi.



Gambar 7.8. Autoencoder.

Tabel 7.4. Autoencoder.

Autoencoder
Bobot acak awal Lakukan hingga kriteria penghentian terpenuhi: Untuk setiap input $\boldsymbol{x}$:

Lakukan pass feed-forward untuk menghitung aktivasi di semua lapisan tersembunyi, kemudian di lapisan output dapatkan output $\boldsymbol{x'}$ Ukur penyimpangan $\boldsymbol{x'}$ dari input $\boldsymbol{x}$ Backpropagate kesalahan melalui jaringan dan lakukan pembaruan bobot

Kita ingin mempelajari beberapa node kecil dalam data tingkat konseptual skala menengah untuk menghasilkan representasi terkompresi, yang entah bagaimana menangkap fitur inti dari input. Algoritma pelatihan untuk autoencoder dirangkum dalam Tabel 7.4.

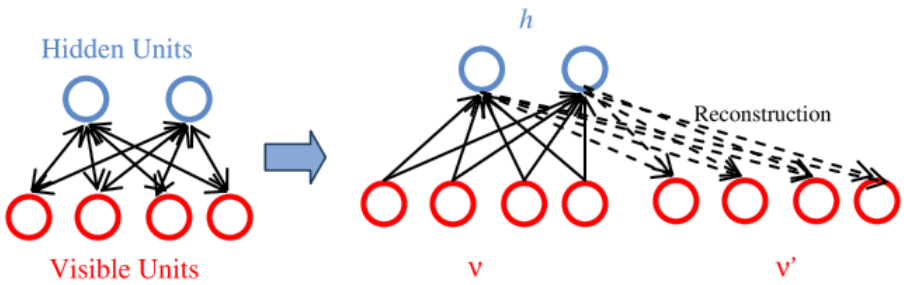
Melatih Restricted Boltzmann Machine

Sebenarnya, ketika RBM diuraikan, ia mirip dengan autoencoder, seperti yang ditunjukkan pada Gambar 7.9. Pelatihan RBM juga mirip dengan pelatihan autoencoder. Pertama, kita menghitung nilai unit tersembunyi h menggunakan vektor dari unit terlihat v dan kemudian merekonstruksi unit terlihat v'. Dengan menggunakan perbedaan antara v dan v', kita dapat memperbarui bobot antara unit terlihat dan unit tersembunyi. Algoritma rinci untuk melatih RBM ditunjukkan pada Tabel 7.5.

Tabel 7.5. Mesin Boltzmann Terbatas.

Mesin Boltzmann terbatas
Bobot acak awal Lakukan hingga kriteria penghentian terpenuhi: Untuk setiap vektor input $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots\}$ Hitung $\boldsymbol{h_j}$ menggunakan $\boldsymbol{X}$, dan nyatakan positif$(\boldsymbol{w_{ij}}) = \boldsymbol{x_i} * \boldsymbol{h_j}$ Rekonstruksi $\boldsymbol{X'}$, dan nyatakan negatif$(\boldsymbol{w_{ij}}) = \boldsymbol{h_j} * \boldsymbol{x_i}$ Lakukan pembaruan bobot: $\boldsymbol{w_{ij}} = \boldsymbol{w_{ij}} + \boldsymbol{\alpha} * (\text{positif}(\boldsymbol{w_{ij}}) - \text{negatif}(\boldsymbol{w_{ij}}))$

Ketika RBM digabungkan lapis demi lapis di mana unit terlihat dari lapisan atas adalah unit tersembunyi dari lapisan bawah, mereka membentuk DBN.



Gambar 7.9. RBM yang dibuka.

7.4 KERANGKA KERJA DEEP LEARNING

Terdapat banyak kerangka kerja deep learning open source dari perusahaan rintisan di bidang ini. Kita akan membahas kerangka kerja ini di bagian ini. Caffe (*Convolutional*

Architecture for Fast Feature Embedding) dikembangkan oleh Berkeley Vision and Learning Center. Ini mungkin merupakan toolkit paling populer dalam visi komputer. Caffe memiliki implementasi jaringan saraf konvolusi yang sangat baik, tetapi dukungan untuk jaringan saraf berulang kurang baik. Caffe menyediakan antarmuka untuk C++, Python, dan Matlab dan dapat dijalankan pada CPU dan GPU.

Tensorflow dikembangkan oleh Google. Ia memiliki API dan implementasi jaringan saraf berulang yang ideal. Tensorflow mendukung komputasi terdistribusi heterogen, dan dapat dijalankan pada berbagai platform, mulai dari ponsel, CPU/GPU tunggal hingga sistem terdistribusi dengan ratusan GPU. Ia mendukung C++ dan Python, tetapi tidak dapat diimplementasikan pada Windows. CNTK dikembangkan oleh Microsoft yang awalnya hanya digunakan untuk asisten pribadi Cortana dari Microsoft dan layanan Skype Translator, dan dirilis pada April 2015. CNTK memungkinkan sistem untuk mengenali ucapan manusia.

Theano lahir di Politeknik Montreal. Banyak paket pembelajaran mendalam yang berasal dari Theano termasuk Blocks dan Keras. Ia menyediakan antarmuka Python dan dapat mendukung sebagian besar model jaringan saraf. Torch telah ada selama lebih dari sepuluh tahun. Ia diimplementasikan pada LuaJIT, yang memiliki kecepatan lebih baik dibandingkan dengan C++, C#, Java. Satu-satunya masalah adalah Lua bukanlah bahasa pemrograman utama. Selain itu, ada banyak kerangka kerja pembelajaran mendalam lainnya seperti Brainstorm yang dapat digunakan untuk menangani jaringan saraf super-dalam dengan ratusan lapisan tersembunyi, Deeplearning4j yang merupakan kerangka kerja pembelajaran mendalam untuk Java, dan MXnet yang menekankan efisiensi penggunaan memori, dan sebagainya.

7.5 PEMBELAJARAN MENDALAM DAN BIG DATA

Di industri, kebanyakan orang berpikir bahwa pembelajaran mesin dangkal lebih efektif untuk menangani big data. Misalnya, dalam banyak aplikasi big data, model linier paling sederhana telah banyak digunakan. Namun, baru-baru ini kemajuan pembelajaran mendalam mendorong kita untuk memikirkan kembali gagasan ini. Dalam kasus big data, mungkin hanya model yang lebih kompleks atau model dengan kemampuan ekspresi tinggi yang dapat sepenuhnya memanfaatkan kekayaan informasi yang tersembunyi dalam jumlah data yang sangat besar. Mungkin kita dapat mempelajari informasi dan pengetahuan yang lebih berharga menggunakan pembelajaran mendalam.

Model dangkal memiliki fitur penting — mengekstrak pola atau fitur dari data sampel berdasarkan pengalaman, dan model dangkal terutama bertanggung jawab untuk klasifikasi atau prediksi. Jadi, kualitas pola yang diekstrak menjadi hambatan dalam kinerja sistem secara keseluruhan. Dengan demikian, umumnya lebih banyak upaya dari tim pengembang dicurahkan untuk menemukan pola yang lebih baik. Untuk menemukan pola yang baik, pengembang perlu memiliki pemahaman mendalam tentang masalah tersebut. Dengan demikian, mendesain pola sampel secara artifisial mungkin bukan pendekatan yang baik.

Di sisi lain, pembelajaran mendalam bertujuan untuk mempelajari fitur yang lebih bermanfaat dari big data melalui pembangunan model mendalam dengan banyak lapisan

tersembunyi dan pelatihan sejumlah besar data. Berbeda dengan pembelajaran mesin dangkal, pembelajaran mendalam menekankan kedalaman model dan pentingnya pelatihan, sehingga dapat meningkatkan akurasi klasifikasi atau prediksi dengan mengekstrak fitur yang lebih tepat. Dibandingkan dengan metode perancangan buatan, pembelajaran mendalam menggunakan pelatihan data besar untuk mempelajari fitur, yang memiliki kemampuan lebih baik untuk mengekspresikan fitur informasi yang kaya. Jadi, dalam beberapa tahun ke depan, kita mungkin akan melihat semakin banyak aplikasi data besar yang menggunakan pembelajaran mendalam sebagai pengganti model dangkal linier.

7.6 APLIKASI DEEP LEARNING

Deep learning merupakan terobosan penting yang dicapai di bidang kecerdasan buatan dalam dekade terakhir, khususnya dalam pengenalan suara, pemrosesan bahasa alami, visi komputer, analisis gambar dan video, serta multimedia. Pada bagian ini, penulis mencantumkan beberapa contoh aplikasi dan merangkum karya mereka.

Pemahaman Gambar

Terobosan paling berpengaruh dalam visi komputer terjadi pada tahun 2012 ketika tim Hinton menggunakan deep learning untuk memenangkan juara klasifikasi gambar ImageNet. Hasil mereka menimbulkan kejutan besar di bidang visi komputer dan membuka jalan bagi deep learning. Dan dalam kompetisi ImageNet ILSVRC 2013, semua 20 tim teratas menggunakan teknologi deep learning. Di antara mereka, pemenangnya adalah Rob Fergus dari Universitas New York (Rob Fergus) dan tingkat kesalahannya adalah 11,197%. Model mereka disebut Clarifai. Pada ILSVRC 2014, pemenangnya GoogLeNet mengurangi tingkat kesalahan menjadi 6,656%.

Deteksi objek adalah salah satu tugas yang lebih sulit daripada klasifikasi objek. Sebuah gambar dapat mencakup sejumlah objek yang termasuk dalam kelas yang berbeda dan deteksi objek adalah untuk menentukan lokasi dan jenis setiap objek. Dalam ImageNet ILSVRC 2013, penyelenggara menambahkan tugas deteksi objek yang mengharuskan untuk mendeteksi objek yang termasuk dalam 200 kelas dalam 40.000 gambar. Pemenang menggunakan fitur desain manual, dan rata-rata presisi (mAP) hanya 22,581%. Dalam ILSVRC 2014, pembelajaran mendalam meningkatkan angka tersebut menjadi 43,933%. Karya berpengaruh lainnya yang menggunakan pembelajaran mendalam untuk visi mesin termasuk RCNN, Overfeat, GoogLeNet, DeepID-Net, jaringan dalam jaringan, VGG, dan pengumpulan piramida spasial dalam CNN mendalam.

Terobosan penting lainnya dalam pembelajaran mendalam pada pengenalan objek adalah pengenalan wajah. Tantangan terbesar pengenalan wajah adalah membedakan antara variasi intra-kelas karena perbedaan cahaya, gerak tubuh, dan ekspresi wajah, dan variasi ekstra-kelas yang disebabkan oleh perbedaan identitas. Distribusi kedua perubahan ini bersifat nonlinier dan sangat kompleks.

Model linier tradisional tidak dapat membedakannya secara efektif. Tujuan pembelajaran mendalam adalah untuk memperoleh representasi fitur baru melalui beberapa lapisan transformasi nonlinier. Fitur-fitur baru ini harus menghilangkan variasi intra-kelas

sebanyak mungkin, sambil mempertahankan variasi ekstra-kelas. Dalam Konferensi Internasional IEEE tentang Visi Komputer dan Pengenalan Pola 2014, DeepID dan DeepFace menggunakan pengenalan wajah sebagai sinyal pengawasan dan mencapai tingkat pengenalan 97,45% dan 97,35% pada basis data wajah LFW (*Labeled Faces in the Wild*).

Pemahaman Ucapan

Pengenalan Ucapan Otomatis (*Automatic Speech Recognition*/ASR), yang bertujuan untuk memungkinkan interaksi manusia-mesin yang alami, telah menjadi area penelitian intensif selama beberapa dekade. Terdapat banyak sistem komersial untuk pengenalan suara seperti AT&T Watson, Microsoft Speech Server, Google Speech API dan Nuance Recognizer.

Sejak diluncurkan pada tahun 2009, transkripsi Google Voice telah menggunakan model akustik *Gaussian Mixture Model* (GMM), yang merupakan teknologi terkini dalam pengenalan suara, selama lebih dari 30 tahun. Teknik-teknik canggih seperti adaptasi model terhadap suara pembicara telah meningkatkan metode pemodelan yang relatif sederhana ini. Kemudian sekitar tahun 2012, *Deep Neural Networks* (DNN) merevolusi bidang pengenalan suara. Jaringan multi-layer ini membedakan suara lebih baik daripada GMM dengan menggunakan "pelatihan diskriminatif," membedakan unit fonetik alih-alih memodelkan masing-masing secara independen. Teknologi telah berkembang pesat dengan Jaringan Neural Berulang (RNN), dan khususnya LSTM RNN yang pertama kali diluncurkan di pengenal suara Android pada Mei 2012. Dibandingkan dengan DNN, LSTM RNN memiliki koneksi berulang tambahan dan sel memori yang memungkinkan mereka untuk "mengingat" data yang telah mereka lihat sejauh ini.

Pemahaman Video

Pembelajaran mendalam untuk klasifikasi aplikasi video masih dalam tahap awal. Model pembelajaran mendalam yang diperoleh dari ImageNet dapat digunakan untuk menggambarkan fitur gambar statis video. Kesulitannya adalah bagaimana menggambarkan karakteristik dinamis. Pendekatan yang paling langsung adalah menganggap video sebagai gambar tiga dimensi, dan langsung menggunakan jaringan neural konvolusi. Tetapi ide ini tampaknya tidak memperhitungkan perbedaan dimensi waktu dan dimensi ruang.

Ide sederhana namun lebih efektif lainnya adalah menghitung terlebih dahulu statistik spasial aliran optik atau fitur dinamis lainnya sebagai input konvolusi. Terdapat juga penelitian tentang penggunaan auto-encoder dalam untuk mengekstrak fitur dinamis secara nonlinier. Dalam penelitian terbaru, *Long Short-Term Memory* (LSTM) menarik perhatian luas, yang merupakan jenis khusus dari jaringan saraf berulang untuk memodelkan ketergantungan rentang variabel yang terdapat dalam tugas pemahaman video.

Pemrosesan Bahasa Alami (NLP)

Selain suara dan gambar, aplikasi lain dari pembelajaran mendalam adalah pemrosesan bahasa alami (NLP). Aplikasi NLP pertama yang menggunakan jaringan saraf adalah model bahasa. Pada tahun 2003, Yoshua Bengio mengusulkan untuk memetakan sebuah kata ke ruang vektor menggunakan embedding, dan kemudian menggunakan jaringan saraf non-linier untuk merepresentasikan model N-Gram. Saat ini, pembelajaran mendalam telah digunakan dalam banyak aplikasi NLP seperti pembuatan teks dan terjemahan.

Secara keseluruhan, kemajuan pembelajaran mendalam pada NLP tidak sehebat pada suara atau gambar. Dibandingkan dengan suara dan gambar, bahasa adalah satu-satunya sinyal buatan yang dihasilkan dan diproses oleh otak manusia. Jadi penulis percaya masih banyak yang dapat dieksplorasi untuk NLP menggunakan pembelajaran mendalam.

Pengenalan Gambar Kompleks

Pengenalan gambar kompleks adalah area yang sulit menggunakan algoritma pembelajaran dangkal yang ada. Di dunia nyata, kita sering harus memproses banyak masalah pengenalan gambar kompleks. Misalnya, bagaimana mengenali tulisan tangan?

Pengenalan Target

Seringkali, kita dapat menemukan beberapa objek, terutama objek yang bergerak dengan kecepatan tinggi, tetapi sangat sulit untuk mengetahui apa objek tersebut. Salah satu alasannya adalah kita hanya dapat mengumpulkan sedikit informasi saat menemukan objek. Oleh karena itu, kita perlu mengembangkan beberapa metode baru untuk mengenali target-target ini. Deep learning mungkin dapat melakukan pekerjaan ini dengan baik di masa depan.

7.7 HAMBATAN DEEP LEARNING

Meskipun deep learning telah membuat kemajuan signifikan dan telah digunakan dalam banyak aplikasi dalam beberapa dekade terakhir, masih ada beberapa kesulitan yang perlu kita pertimbangkan dan selesaikan di masa depan.

Kesulitan Teoretis

Meskipun deep learning telah mencapai kesuksesan besar dalam praktik, masih banyak pekerjaan yang harus dilakukan pada analisis teoretis. Misalnya, kapan jaringan akan konvergen? Bagaimana cara mencapai fitur lokal yang lebih baik? Invariansi apa yang diekstrak dari transformasi di setiap lapisan dan informasi apa yang hilang dalam prosedur ini? Karena model deep learning bersifat non-konveks, penelitian teoretis di bidang ini mungkin sulit. Baru-baru ini Mallat melakukan analisis kuantitatif untuk jaringan deep learning menggunakan wavelet, yang mungkin merupakan arah yang penting.

Kesulitan Pemodelan

Karena analisis teoretis sangat sulit untuk model deep learning saat ini, apakah mungkin untuk merancang model baru yang memiliki kemampuan belajar yang sama kuatnya dengan model yang ada dan pada saat yang sama lebih mudah dianalisis? Selain itu, seperti yang telah penulis sebutkan di subBab sebelumnya, arsitektur yang berbeda hanya cocok untuk aplikasi yang berbeda. Jadi pertanyaan lain adalah bagaimana merancang model terpadu baru yang dapat menangani berbagai jenis data seperti bahasa, ucapan, dan gambar? Itulah pertanyaan-pertanyaan yang perlu kita pikirkan sambil mempromosikan studi analisis teoretis dan kemampuan komputasi model deep learning saat ini.

Kesulitan Industri

Bagi perusahaan internet, bagaimana memanfaatkan platform komputasi paralel masif untuk melatih big data adalah masalah terpenting saat menggunakan teknologi deep learning. Platform big data tradisional seperti Hadoop tidak cocok untuk komputasi deep learning dengan iterasi yang sering karena latensinya yang tinggi. Teknik pelatihan DNN yang ada

sebagian besar menggunakan metode gradien stokastik (SGD) sebagai metode pelatihan. Metode ini tidak mungkin diparalelkan di antara banyak komputer.

Bahkan pelatihan dengan GPU, proses pembelajaran DNN tradisional masih akan memakan waktu yang sangat lama. Dengan meningkatnya jumlah big data di internet, pelatihan dengan kecepatan lambat ini tidak mampu memenuhi kebutuhan aplikasi internet. Google DistBelief adalah platform komputasi paralel untuk pembelajaran model mendalam menggunakan algoritma asinkron. Dengan perkembangan teknologi pelatihan data besar-besaran, akurasi pengenalan suara dan gambar dapat ditingkatkan lebih lanjut di masa mendatang.

BAB 8
JARINGAN KONVOLUSIONAL UNTUK PENGENALAN KONSEP VISUAL

Pengenalan konsep visual merupakan bidang penelitian aktif dalam dekade terakhir. Terkait dengan perhatian ini, arsitektur pembelajaran mendalam menunjukkan potensi besar di berbagai domain visi komputer termasuk klasifikasi gambar, deteksi objek, deteksi peristiwa, dan pengenalan aksi dalam video. Dalam bab ini, penulis menyelidiki berbagai aspek jaringan saraf konvolusional untuk pengenalan konsep visual. Penulis menganalisis studi terbaru dan berbagai arsitektur jaringan baik dari segi waktu eksekusi maupun akurasi.

Dalam sistem pengenalan konsep visual yang penulis usulkan, pertama-tama penulis membahas berbagai properti penting dari arsitektur jaringan konvolusional populer yang sedang dipertimbangkan. Kemudian penulis menjelaskan metode penulis untuk ekstraksi fitur pada berbagai tingkat abstraksi. Penulis menyajikan informasi empiris yang ekstensif beserta praktik terbaik untuk praktisi big data. Dengan menggunakan praktik terbaik ini, penulis mengusulkan mekanisme fusi yang efisien baik untuk model jaringan tunggal maupun ganda. Penulis menyajikan hasil terkini pada dataset benchmark sambil menjaga biaya komputasi tetap rendah. Hasil penulis menunjukkan bahwa hasil terkini ini dapat dicapai tanpa menggunakan teknik augmentasi data yang ekstensif.

8.1 PERKEMBANGAN DAN KEUNGGULAN CNN

Dalam beberapa tahun terakhir, jaringan saraf konvolusional (CNN) telah menunjukkan kinerja yang sangat baik dalam berbagai aplikasi visi komputer. Sebagian besar keberhasilan ini terutama disebabkan oleh dua faktor; ketersediaan arsitektur pemrosesan paralel baru-baru ini dan kumpulan data gambar yang lebih besar. Kedatangan kumpulan data beranotasi dan berukuran lebih besar seperti ILSVRC, LabelMe dan MIT Places, memungkinkan untuk melatih jaringan konvolusional dalam untuk tugas yang ada tanpa mengalami hambatan overfitting. Lebih lanjut, aksesibilitas sumber daya komputasi yang lebih tinggi memungkinkan para peneliti untuk merancang jaringan yang lebih dalam dengan lebih banyak parameter. Misalnya, jaringan berkinerja terbaik dalam tantangan ILSVRC 2014 menggunakan lebih dari 100 juta parameter. Perpaduan kedua faktor ini dengan algoritma pelatihan yang lebih baik menghasilkan arsitektur yang sangat sukses yang menghasilkan hasil terbaik di hampir setiap bidang visi komputer.

Salah satu properti yang sangat penting dari CNN adalah kemampuan generalisasinya. Dimungkinkan untuk menerapkan jaringan yang telah dilatih sebelumnya ke dataset atau domain aplikasi yang sama sekali berbeda dan mencapai hasil yang mendekati kondisi terkini. Terlebih lagi, dengan upaya pelatihan yang sangat sedikit, mereka dapat mengalahkan sebagian besar pendekatan terkini lainnya. Hal ini menimbulkan pertanyaan apakah mereka dapat diterapkan pada domain aplikasi di mana keragaman data merupakan tantangan. Studi terbaru memang menunjukkan bahwa hal ini sangat mungkin.

Selain kemampuan transferabilitasnya, CNN cocok untuk paralelisasi agresif baik di lingkungan terdistribusi maupun di lingkungan komputasi GPU berkinerja tinggi. Setelah dilatih, mereka dapat mencapai kinerja waktu eksekusi yang lebih tinggi daripada waktu nyata selama inferensi. Sifat-sifat menarik dari CNN ini menjadikannya kandidat yang kuat untuk aplikasi multimedia besar. Apakah mereka dapat menjadi lebih dari sekadar kandidat dan menjadi bagian dari aplikasi yang menuntut bergantung pada bagaimana kinerja mereka dengan batas baru big data, yaitu video. Pertanyaan ini juga merupakan salah satu motivasi penulis untuk menyelidiki lebih lanjut kinerja CNN untuk aplikasi big data. Upaya penulis adalah untuk mencapai implementasi yang efisien dalam biaya komputasi sambil mempertahankan akurasi terbaik untuk tugas yang ada.

Ketersediaan arsitektur GPU paralel masif memungkinkan para peneliti untuk mencoba berbagai opsi konfigurasi. Salah satu teknik yang paling umum adalah penggabungan berbagai pendekatan. Wang dkk. menunjukkan bagaimana beberapa arsitektur CNN yang dilatih pada dataset yang berbeda dapat digunakan untuk meningkatkan akurasi pengenalan peristiwa dalam gambar. Guo dan Gould menerapkan penggabungan jaringan yang berbeda untuk deteksi objek. Chatfield dkk. menunjukkan ensemble arsitektur CNN dengan vektor Fisher, meskipun peningkatannya marginal.

Seperti yang telah ditunjukkan dalam banyak penelitian sebelumnya, detail implementasi memainkan peran penting dalam kinerja algoritma visi. Untuk memiliki pemahaman yang lebih baik tentang kinerja relatif dari berbagai implementasi, yang terbaik adalah membandingkan alternatif secara sistematis menggunakan pipeline pemrosesan yang serupa, jika tidak identik.

Dalam bab ini, penulis membandingkan implementasi CNN canggih terbaru dengan pipeline pemrosesan serupa dengan mempertimbangkan big data. Penulis bertujuan untuk mendapatkan serangkaian praktik terbaik baik mengenai biaya komputasi maupun batasan akurasi menuju penerapan aplikasi analitik big data yang lebih baik. Selain itu, penulis menganalisis kinerja empat arsitektur CNN populer yang berbeda yang diperkenalkan oleh penelitian sebelumnya. Penulis juga menyelidiki efek strategi penggabungan deskriptor untuk ekstraksi fitur yang efisien.

8.2 EVOLUSI METODE PENGENALAN VISUAL

Fitur gambar lokal telah digunakan secara luas untuk klasifikasi gambar selama lebih dari satu dekade. Pada bagian ini, kami secara singkat meninjau tiga pendekatan utama dalam pengenalan konsep visual: *bag-of-words* (BoW), *deep learning*, dan metode berbasis fusi. Csurka menunjukkan cara menggunakan pengodean BoW dengan fitur gambar lokal. Sivic dan Zisserman telah memberikan rincian pengindeksan video menggunakan paradigma BoW. Lazebnik telah meningkatkan BoW dengan menggunakan piramida spasial. Yang lebih lanjut meningkatkan paradigma BoW menggunakan *sparse coding*. Chatfield dan rekan-rekannya telah menunjukkan detail implementasi terbaik untuk berbagai varian pengodean BoW. Daftar ini dapat diperluas dengan banyak studi penting lainnya dari tahun 2000-an, termasuk namun tidak terbatas pada [2, 6–8, 19, 44, 45, 54].

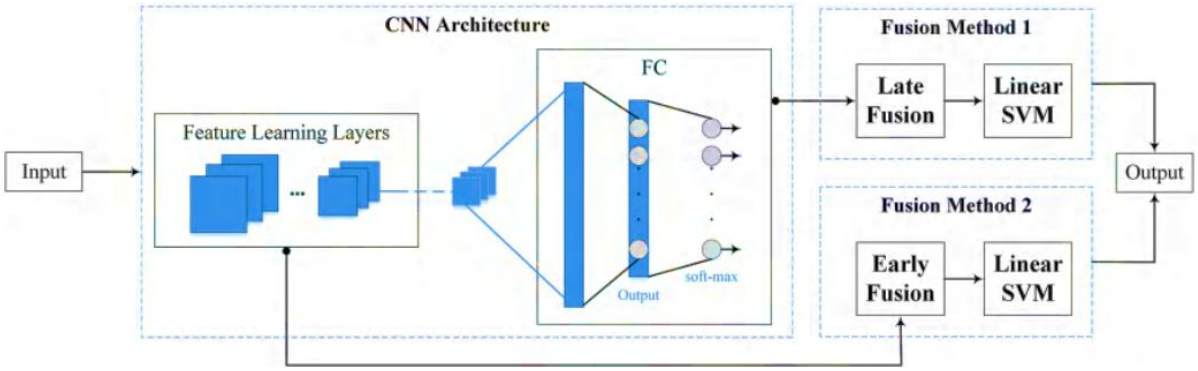
Dengan munculnya performa jaringan saraf tiruan, pendekatan berbasis *deep learning* mulai menggantikan implementasi berbasis BoW. Krizhevsky menerapkan CNN untuk tugas klasifikasi gambar dan menunjukkan hasil yang sangat baik. Zeiler dan Fergus telah menunjukkan kesuksesan di balik jaringan saraf. Mereka juga telah mendemonstrasikan bagaimana model CNN dapat dikembangkan lebih lanjut menggunakan teknik visualisasi. Sermanet dkk. mendorong batas lebih jauh dengan arsitektur CNN mereka dan telah menunjukkan bahwa CNN dapat dilatih untuk memenuhi berbagai tugas gambar seperti klasifikasi dan lokalisasi secara bersamaan. Szegedy memperkenalkan jaringan yang jauh lebih dalam, sementara Simonyan dan Zisserman telah menunjukkan kinerja yang sangat baik dengan arsitektur CNN dalam mereka. Pada tahun-tahun berikutnya, arsitektur yang lebih dalam muncul, yang semakin mendorong akurasi klasifikasi. Pemenang tantangan klasifikasi dan lokalisasi ILSVRC 2014 menggunakan arsitektur yang lebih dalam. Studi terbaru mengungkapkan bahwa arsitektur yang jauh lebih dalam diperkirakan akan muncul dalam waktu yang sangat dekat.

Penerapan arsitektur CNN pada domain video juga sangat luas. Karpathy dkk. menunjukkan bagaimana model CNN dapat diperkaya dengan informasi temporal yang ada dalam video menggunakan beberapa teknik fusi. Simonyan meningkatkan pendekatan umum untuk pengenalan aksi video menggunakan jaringan konvolusional dua aliran (*two-stream convolutional networks*). Ye menganalisis lebih lanjut arsitektur dua aliran untuk klasifikasi video. Wang menguraikan praktik-praktik yang baik untuk pengenalan aksi dalam video. Zha dkk. menerapkan CNN bersama dengan arsitektur dangkal untuk klasifikasi video.

Selain mempelajari arsitektur dan metode pengodean, banyak peneliti telah berupaya meningkatkan kinerja klasifikasi dengan bantuan strategi fusi. Snoek dkk. telah memperkenalkan perbedaan antara fusi awal (*early fusion*) dan fusi akhir (*late fusion*). Sun dkk. telah menggunakan *Canonical Correlation Analysis* (CCA) untuk fusi berbagai modalitas. Yang memilih pendekatan hierarkis untuk fusi fitur. Dalam studi ini, tujuan kami adalah untuk bereksperimen dengan arsitektur CNN dalam konteks fusi dengan menyediakan serangkaian praktik terbaik, baik mengenai biaya komputasi maupun akurasi pengenalan konsep visual.

8.3 JARINGAN NEURAL KONVOLUSIONAL DALAM

Pada bagian ini, penulis memperkenalkan tiga komponen kunci dari sistem kami. Pertama, penulis menjelaskan properti penting dari arsitektur CNN dan merinci berbagai arsitektur yang penulis gunakan dalam penelitian kami. Selanjutnya, penulis membahas cara mengekstrak fitur menggunakan arsitektur CNN dan menjelaskan metode penulis untuk ekstraksi fitur menggunakan peta fitur dari lapisan konvolusional yang diberikan. Kemudian, penulis memperkenalkan skema fusi sederhana namun inovatif penulis untuk mengatasi daya representatif dan kinerja waktu eksekusi dari arsitektur CNN yang berbeda menggunakan berbagai tingkat abstraksi. Gambar 8.1 mengilustrasikan sistem pengenalan keseluruhan kami.



Gambar 8.1. Ilustrasi sistem yang yang dibangun

Dalam sistem yang penulis usulkan, penulis mengekstrak fitur gambar dari beberapa lapisan berbeda dari arsitektur CNN yang diberikan. Detail ekstraksi fitur tersebut dapat ditemukan di bagian berikutnya. Dalam skema fusi awal, semua fitur yang diekstrak digabungkan dengan metode fusi yang tepat sebelum dimasukkan ke SVM.

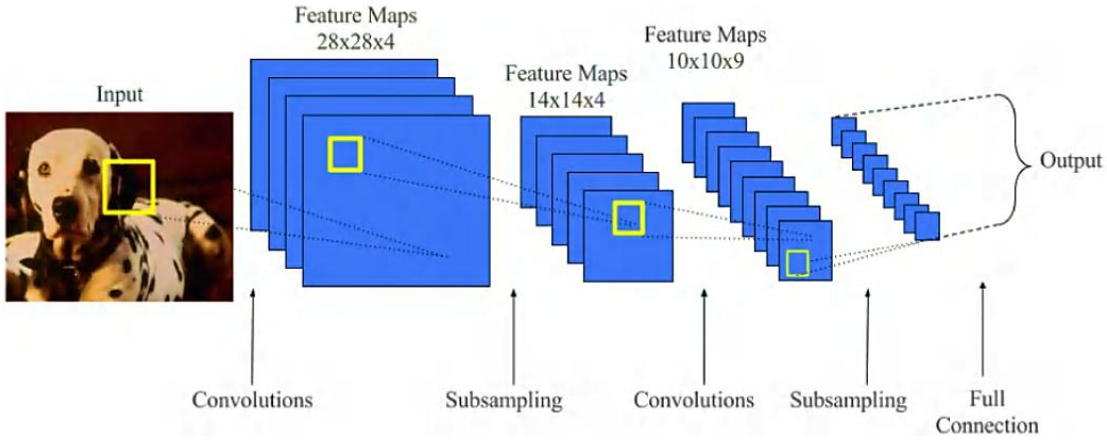
Dalam kasus beberapa arsitektur CNN, beberapa fitur dari arsitektur CNN yang berbeda juga digabungkan sebelum klasifikasi SVM. Dalam kasus fusi akhir, penulis melatih pengklasifikasi terpisah untuk setiap lapisan fitur dari setiap arsitektur CNN dan menggabungkan hasil dari pengklasifikasi ini menggunakan pemungutan suara sederhana untuk mencapai skor klasifikasi tunggal.

Jaringan Neural Konvolusional

Arsitektur Jaringan Neural Konvolusional (CNN) adalah desain khusus dari jaringan neural buatan dengan tata letak lapisan dan paradigma koneksi tertentu. Arsitektur CNN dapat berisi beberapa lapisan konvolusional, pooling, dan lapisan yang terhubung sepenuhnya. Konsensus umum dalam literatur adalah hanya menghitung lapisan konvolusional dan lapisan yang terhubung sepenuhnya dalam representasi mendalam akhir dari jaringan tertentu, dengan kata lain jumlah lapisan pooling tidak berkontribusi pada kedalaman arsitektur CNN tertentu. Struktur umum arsitektur CNN digambarkan pada Gambar 8.2.

Lapisan konvolusional tidak lain adalah operasi konvolusi gambar dari lapisan sebelumnya. Ia mengambil sub-wilayah persegi panjang tertentu dari lapisan sebelumnya dan menerapkan operasi konvolusi ke sub-wilayah ini. Prosedur ini diterapkan pada skema jendela geser pada lapisan sebelumnya di kedua dimensi spasial sehingga output dari operasi ini adalah representasi persegi panjang lain yang disebut sebagai peta fitur. Lapisan konvolusional tertentu umumnya berisi beberapa peta fitur ini. Karena peta fitur yang berbeda akan memiliki koefisien konvolusi yang berbeda setelah pelatihan, mereka dikatakan mempelajari, atau mencerminkan, karakteristik yang berbeda dari input tertentu.

Dengan kata lain, mereka dapat dianggap sebagai pengklasifikasi linier pada patch data yang diberikan di mana setiap peta fitur, atau filter konvolusi, sesuai dengan representasi tingkat menengah tertentu. Peta fitur dalam lapisan konvolusi dapat dihubungkan ke semua peta fitur di lapisan sebelumnya sementara filter lapisan konvolusi pertama, atau peta fitur, diberi input berupa gambar yang ada.



Gambar 8.2. Diagram blok arsitektur CNN tipikal.

Jumlah lapisan konvolusi, karakteristik koneksinya, lapisan pooling yang mendahului dan mengikuti lapisan tersebut, jumlah peta fitur di semua lapisan konvolusi dan parameter konvolusinya seperti ukuran kernel, stride, padding semuanya ditetapkan berdasarkan desain untuk arsitektur CNN tertentu. Kita harus menekankan bahwa desain lapisan konvolusi tidak memaksakan batasan apa pun pada resolusi gambar input yang diberikan, yang merupakan masalah umum pada arsitektur CNN yang akan kita sebutkan di paragraf berikutnya.

Di sisi lain, lapisan yang terhubung sepenuhnya (fully-connected layers) sebagian besar seperti perceptron multi-layer di mana semua neuron di lapisan sebelumnya, baik itu konvolusional/pooling atau terhubung sepenuhnya, terhubung ke setiap neuron yang dimilikinya. Lapisan yang terhubung sepenuhnya bertindak sebagai lapisan perataan data sehingga outputnya dapat dianggap sebagai vektor fitur 1-dimensi, tidak seperti lapisan konvolusional.

Di sisi lain, informasi spasial yang ada di lapisan konvolusional hilang di lapisan yang terhubung sepenuhnya. Implikasi dari perilaku struktural ini adalah bahwa tidak boleh ada lagi lapisan konvolusional setelah lapisan yang terhubung sepenuhnya dan resolusi gambar input dari arsitektur CNN kita harus memiliki ukuran yang telah ditentukan sebelumnya. Karena sifat lapisan yang terhubung sepenuhnya ini, outputnya lebih seperti deskriptor gambar umum daripada fitur lokal. Kerugian lain yang ditimbulkan oleh lapisan yang terhubung sepenuhnya adalah kita juga harus memproses gambar yang ada terlebih dahulu untuk memiliki resolusi tetap sebelum melewatinya melalui jaringan kita.

Berdasarkan definisi arsitektur CNN ini, dapat dikatakan bahwa peta fitur lapisan konvolusional dapat dianggap sebagai fitur gambar lokal dan tidak dapat dihindari untuk membandingkannya dengan fitur gambar lokal buatan tangan seperti SIFT atau SURF. Lebih lanjut, lapisan yang terhubung sepenuhnya dapat dianggap sebagai pengkodean atau representasi dari fitur gambar lokal ini sehingga dapat dibandingkan dengan algoritma pengkodean fitur gambar sintetis seperti representasi *bag-of-words* (BoW), *Vector of Locally Aggregated Descriptors* (VLAD), dan *Fisher Vectors* (FV).

Berdasarkan hubungan ini, dapat dikatakan bahwa, tidak seperti pendekatan rekayasa yang terdiri dari fitur lokal buatan tangan yang diikuti oleh pengkodean khusus, CNN dapat

mempelajari representasi gambar lokal dan metodologi pengkodean yang sesuai untuk tugas pemrosesan gambar yang sedang dikerjakan. Meskipun keunggulan arsitektur CNN dibandingkan metode rekayasa telah terbukti untuk sebagian besar tugas pemrosesan gambar, apakah kombinasi keduanya dapat meningkatkan situasi merupakan area penelitian yang sedang berlangsung.

Arsitektur CNN yang Digunakan

Penulis menyelidiki kinerja waktu eksekusi dari empat arsitektur CNN populer. Pertama adalah karya inovatif Krizhevsky dkk. yang diperkenalkan pada tahun 2012 untuk tugas klasifikasi gambar dari tantangan ILSVRC. Arsitektur mereka terdiri dari delapan lapisan, lima di antaranya adalah lapisan konvolusional. Tiga lapisan terakhir dari jaringan adalah lapisan yang terhubung sepenuhnya. Lapisan terakhir adalah lapisan klasifikasi softmax yang secara khusus disetel untuk 1000 kelas ImageNet ILSVRC.

Model CNN kedua yang dimaksud adalah pemenang tantangan ILSVRC 2014. Jaringan mereka adalah arsitektur yang disebut “Jaringan-dalam-Jaringan”. Arsitektur mereka terdiri dari 22 lapisan tanpa lapisan pooling. Arsitektur mereka berisi sekitar 100 lapisan secara keseluruhan jika blok bangunan independen dihitung sebagai lapisan terpisah. Perhatian utama mereka adalah mampu memanfaatkan jaringan yang lebih besar sambil menjaga kompleksitas komputasi pada tingkat yang lebih rendah. Pendekatan mereka adalah untuk menjaga sparsitas tingkat filter sambil menghindari inefisiensi perangkat keras komputasi saat ini.

Model ketiga yang baru saja adalah runner-up tantangan klasifikasi ILSVRC 2014 dan pemenang tugas lokalisasi tahun yang sama. Dua jaringan konvolusi mereka sangat mirip dalam desain tetapi satu lebih dalam 3 lapisan daripada yang lain. Jika dibandingkan dengan arsitektur jaringan lain, mereka menggunakan filter konvolusi yang lebih kecil di semua lapisan jaringan yang memungkinkan mereka untuk meningkatkan kedalaman jaringan mereka. Penulis mengevaluasi arsitektur jaringan mereka yang lebih dalam yang terdiri dari 19 lapisan dan disebut VGG19 dalam berbagai publikasi.

Model keempat yang penulis pertimbangkan adalah pemenang tantangan ILSVRC terbaru (2015). Arsitektur ini memiliki kedalaman lebih dari 100 lapisan dan tiga versi jaringan pra-terlatih tersedia secara publik. Penulis menggunakan versi jaringan terdalam dengan total 152 lapisan dan menunjuk arsitektur ini sebagai ResNet152 dalam penelitian kami.

Mengekstraksi Fitur CNN

Ada beberapa cara untuk mengekstrak fitur gambar, atau bingkai video, menggunakan arsitektur CNN. Metode pertama dan terpenting yang digunakan adalah menerapkan pemrosesan arsitektur CNN kita ke gambar tertentu dan langsung mengekspor aktivasi pada salah satu lapisan yang terhubung sepenuhnya sebagai fitur gambar global akhir kita, yaitu deskriptor gambar.

Metode ini hanya bergantung pada kinerja representatif arsitektur CNN kita untuk gambar yang ada karena kita menggunakan informasi yang ada dalam peta fitur yang dipelajari dan pengkodean yang dipelajari pada lapisan yang terhubung sepenuhnya. Menggunakan arsitektur CNN sebagai deskriptor gambar sederhana dengan cara ini secara mengejutkan

kompetitif atau lebih unggul daripada metodologi yang ada untuk berbagai tugas pemrosesan gambar. Penulis menggunakan skema ekstraksi fitur ini sebagai salah satu skema ekstraksi fitur dalam penelitian kami. Penulis mengevaluasi kinerja beberapa lapisan yang terhubung sepenuhnya dari arsitektur CNN di bagian eksperimen kami.

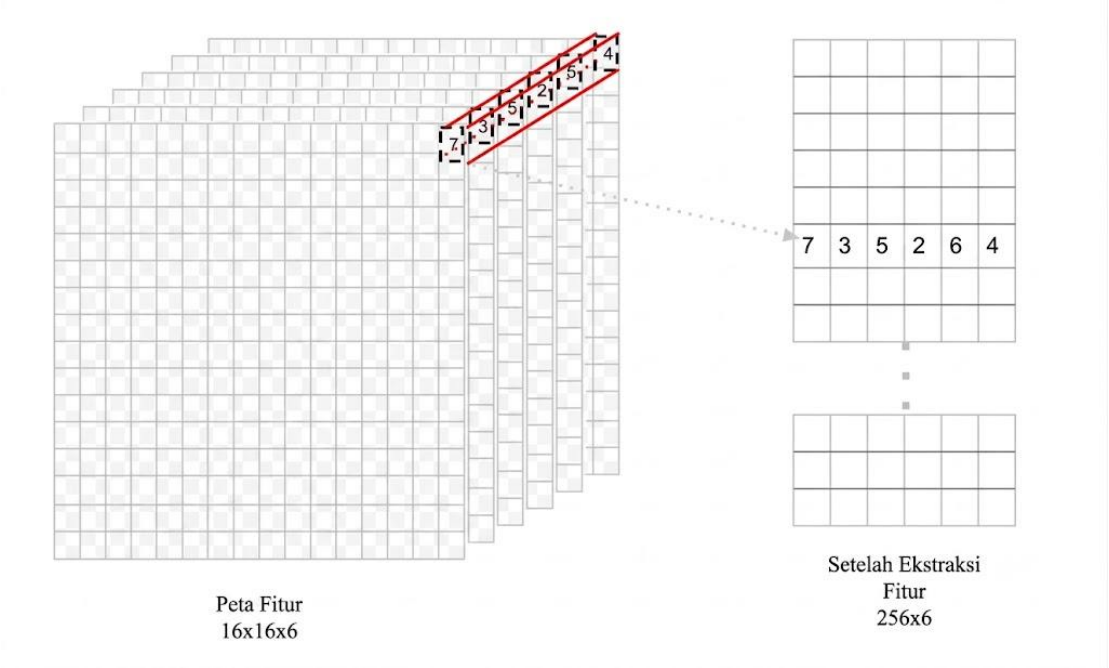
Metode ekstraksi fitur lainnya adalah dengan menganggap arsitektur CNN sebagai ekstraktor fitur lokal dan membuang informasi yang ada di lapisan yang terhubung sepenuhnya. Dalam metode ini, peta fitur konvolusional diperlakukan sebagai fitur lokal dan fitur lokal ini dienkode, atau diagregasi, menggunakan metode yang sesuai seperti pengkodean bag-of-words, pengkodean VLAD, atau pengkodean Fisher Vector.

Selain pengkodean tingkat tinggi ini, Babenko dan Lempitsky telah menyarankan untuk menggunakan sum-pooling sebagai metode agregasi fitur lokal yang lebih sederhana namun efektif dan mereka telah mengevaluasi proposal mereka untuk tugas pengambilan gambar. Sum-pooling secara komputasi jauh lebih sederhana dan non-parametrik dibandingkan dengan pengkodean tingkat tinggi. Kami mengikuti penalaran mereka dalam pembelajaran ini dan menggunakan sum-pooling sebagai metode ekstraksi fitur utama kami untuk lapisan konvolusional. Selain sum-pooling, kami juga mengevaluasi kinerja max-pooling dan membandingkan hasilnya dengan sum-pooling. Kami tidak menggunakan penggabungan (*concatenation*) sebagai alternatif di sini karena kardinalitas fitur yang tinggi pada lapisan konvolusi yang akan menghasilkan terlalu banyak overhead kinerja pada pengklasifikasi baik dalam hal memori maupun penggunaan prosesor. Kami membandingkan kardinalitas fitur untuk kasus penggabungan dan pooling pada Tabel 8.1.

Metode ekstraksi fitur kami menggunakan peta fitur untuk lapisan konvolusi tertentu digambarkan pada Gambar 8.3. Dalam pengaturan ini, kami mengasumsikan peta fitur kami memiliki lebar W dan tinggi H dan misalkan ada M peta fitur yang ada di lapisan konvolusi tertentu. Melewatkan gambar melalui lapisan ini menghasilkan himpunan fitur lokal $X = \{x_1, x_2, \dots, x_n\}$ di mana x_i adalah vektor fitur yang diekstrak pada satu posisi spasial dari semua peta fitur. Ini berarti x_i memiliki dimensi M dan jumlah fitur yang ada dalam himpunan X adalah $n = W * H$. Setelah konversi peta fitur menjadi vektor fitur ini, operasi fusi lebih lanjut akan diterapkan dalam domain fitur, bukan domain spasial.

Tabel 8.1 Dimensi lapisan konvolusi dari berbagai arsitektur.

Nama lapisan	Jaringan	Dimensi	Ukuran linear	Ukuran maks/jumlah
conv1	AlexNet	55 x 55 x 96	290400	96
conv2	AlexNet	27 x 27 x 256	186624	256
conv4-1	VGG19	28 x 28 x 512	401408	512
conv5-1	VGG19	14 x 14 x 512	100352	512
inception-5a/output	GoogleNet	7 x 7 x 832	40768	832
inception-5b/output	GoogleNet	7 x 7 x 1024	50176	1024



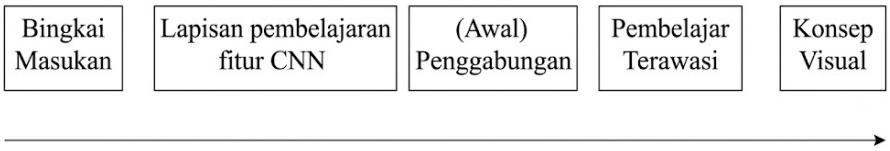
Gambar 8.3 Ekstraksi fitur menggunakan peta fitur dari lapisan konvolusi tertentu.

Skema Fusi yang Diusulkan

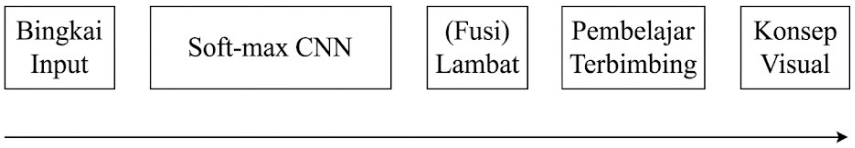
Fusi informasi merupakan elemen penting dari sebagian besar sistem pengambilan/klasifikasi gambar/video di mana peningkatan akurasi sistem secara keseluruhan merupakan langkah maju yang kritis. Menggabungkan informasi yang berbeda tidak hanya penting untuk meningkatkan akurasi tetapi juga untuk mencapai ketahanan yang tinggi terhadap dinamika sistem yang bervariasi. Fusi fitur, atau data, merupakan area pembelajaran yang sedang berlangsung selama lebih dari satu dekade dan masih merupakan masalah terbuka. Ini adalah topik umum yang tidak hanya diteliti oleh peneliti visi komputer tetapi juga menarik perhatian konstan dari area pembelajaran lain.

Dimungkinkan untuk mengkategorikan fusi fitur dalam berbagai aspek tetapi salah satu kategorisasi tahap pertama yang paling diterima adalah membagi fusi fitur menjadi dua kategori: fusi awal dan fusi akhir. Fusi awal, yang juga disebut fusi fitur dalam literatur, mengintegrasikan berbagai sumber informasi sebelum diproses oleh pengklasifikasi target. Dengan kata lain, fitur yang berasal dari berbagai sumber digabungkan menjadi satu representasi.

Dapat dikatakan bahwa penggabungan fitur mendahului pembelajaran konsep dalam jenis penggabungan fitur ini. Sebaliknya, pembelajaran konsep mendahului penggabungan fitur dalam skema penggabungan akhir. Kami menggunakan skema penggabungan awal dan akhir dengan cara yang berbeda bersama dengan arsitektur CNN. Alur pemrosesan skema penggabungan awal dan akhir kami digambarkan pada Gambar 8.4 dan 8.5, masing-masing.



Gambar 8.4 Alur pemrosesan skema fusi awal kami.



Gambar 8.5 Alur pemrosesan skema fusi akhir kami.

Fusi Awal

Strategi fusi awal memiliki beberapa keunggulan dibandingkan strategi fusi akhir asalkan dilakukan dengan benar. Vektor fitur yang berbeda menunjukkan karakteristik yang berbeda dari pola yang sama dan menggabungkan fitur-fitur tersebut mempertahankan informasi diskriminatif yang efektif sambil menghilangkan informasi yang berlebihan, sampai batas tertentu. Misalkan kita memiliki himpunan fitur $F = \{f_1, f_2, \dots, f_n\}$ di mana setiap fitur adalah vektor fitur berdimensi tinggi yang direpresentasikan dengan $f_i \in R^{\mathbb{d}_i}$. Kita harus menekankan bahwa setiap fitur yang berbeda mungkin memiliki kardinalitas yang berbeda sehingga $d_i = \{d_1, d_2, \dots, d_n\}$. Maka fungsi fusi fitur (awal) ψ dapat didefinisikan sebagai operator pemetaan pada F sedemikian sehingga $\psi(F) \rightarrow R^{\mathbb{d}}$. Ketika membandingkan strategi fusi yang berbeda, biasanya operator fusi yang berbeda direpresentasikan dengan sub-simbol yang berbeda seperti 1 dan 2. Kami menggunakan strategi fusi dan operator fusi secara bergantian di seluruh artikel ini.

Strategi fusi awal pertama yang kami selidiki adalah metode penggabungan (*concatenation*) di mana vektor fitur berbeda dari berbagai sumber digabungkan menjadi satu supervektor untuk merepresentasikan fitur gambar akhir. Ini adalah salah satu metode fusi fitur yang paling banyak digunakan dalam literatur dan popularitasnya mungkin disebabkan oleh implementasinya yang sederhana dan kinerja yang relatif baik. Dalam hal ini, kita mengubah himpunan fitur F menjadi supervektor $f_f = [f_1, f_2, \dots, f_n]$ dengan menggabungkan fitur. Ukuran vektor akhir adalah jumlah dari semua dimensi fitur, yaitu $d = \sum_{i=1}^{n} d_i$. Salah satu kelemahan dari pendekatan ini adalah bahwa penggabungan jelas memanfaatkan dimensi fitur akhir dan menyerahkan tugas yang menantang dari diskriminasi fitur sepenuhnya kepada algoritma pembelajaran.

Oleh karena itu, kami menggunakan strategi fusi lain, yaitu perataan (*averaging*), yang juga disebut *summation pooling* atau *sum pooling*, dalam konteks jaringan saraf. Dalam perataan atau sum pooling, kita merata-ratakan himpunan fitur F untuk membentuk deskriptor gambar akhir sebagai $f_f = \frac{1}{n}\sum_{i=1}^{n} f_i$. Dalam metode fusi ini, semua fitur yang ada

dalam himpunan fitur F harus memiliki kardinalitas fitur yang sama atau beberapa bentuk normalisasi dimensi fitur, atau pra-pemrosesan, harus dilakukan sebelum operasi fusi.

Penggabungan maksimum (*maximum pooling*) adalah strategi fusi lain yang kami gunakan. Strategi ini mirip dengan perataan dalam hal persyaratan pra-pemrosesan dan kardinalitas fitur akhir, namun, strategi ini berbeda dalam cara fitur digabungkan. Alih-alih mengambil rata-rata semua elemen fitur, penggabungan maksimum memilih nilai tertinggi dari fitur yang sesuai. Jika representasi fitur akhir kita adalah $f_f \rightarrow R^{\mathbb{d}}$ maka penggabungan maksimum menetapkan setiap anggota f_f sebagai $f_f^i = \max_{i=1}^{d}(f_1^i, f_2^i, \dots, f_n^i)$.

Operasi penggabungan, penggabungan penjumlahan (*summation pooling*), dan penggabungan maksimum (*maximum pooling*) dapat disebut sebagai metode fusi trivial. Deskripsi operasi ini dirangkum dalam Tabel 8.2. Selain operasi fusi trivial ini, kami juga menggunakan mekanisme fusi berdasarkan Pemeriksaan Korelasi Kanonik (CCA) dalam arsitektur CNN. Ini adalah ukuran statistik untuk menemukan hubungan linier antara dua variabel acak. Ini pertama kali diperkenalkan oleh H. Hotelling pada tahun 1936 dan telah digunakan dalam berbagai bidang pembelajaran, dan kami mengadopsi metode ini di sini untuk arsitektur CNN dalam skema fusi awal kami. Misalkan X dan Y adalah dua variabel acak dengan rata-rata nol sedemikian sehingga $X \in R^{d_{\mathbb{1}}}$ dan $Y \in R^{d_{\mathbb{2}}}$ yang masing-masing memiliki matriks kovariansi dalam himpunan sebagai C_{xx} dan C_{yy}. Mari kita juga definisikan matriks kovariansi antar himpunan C_{xy} dan C_{yx}. Maka matriks kovariansi totalnya C dapat didefinisikan sebagai:

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} \tag{8.1}$$

Tabel 8.2 Operasi Fusi Awal yang Sederhana. Kami berasumsi bahwa dimensi fitur dinormalisasi sebelum operasi maksimum dan penjumlahan.

Operasi	Definisi
Menggabungkan	$f_f = [f_1, f_2, \dots, f_n]$
MAX	$f_f \rightarrow R^d$ kemudian, $f_f^i = \max_{i=1}^{d}(f_1^i, f_2^i, \dots, f_n^i)$
SUM	$f_f = \sum_{i=1}^{n} f_i$

Kemudian dimungkinkan untuk menemukan korelasi kanonik antara X dan Y dengan menyelesaikan persamaan nilai eigen (8.2) dan (8.3):

$$C_{xx}^{-1} C_{xy} C_{yy}^{-1} C_{yx} W_x = \rho^2 W_x \tag{8.2}$$

$$C_{yy}^{-1} C_{yx} C_{xx}^{-1} C_{xy} W_y = \rho^2 W_y \tag{8.3}$$

W_x dan W_y adalah vektor eigen dan ρ^2 adalah nilai eigen yang masing-masing juga dikenal sebagai vektor basis korelasi kanonik yang dinormalisasi dan korelasi kanonik kuadrat. Perlu dicatat bahwa jumlah solusi bukan nol untuk Persamaan (8.2) dan (8.3) dibatasi oleh dimensi

terkecil dari variabel acak, yaitu min$(d1, d2)$. Kemudian kita dapat mendefinisikan X* dan Y* sebagai proyeksi ke W_x dan W_x sebagai $X_* = x^T W x$ dan $Y_* = y^T W_y \cdot X^*$ dan Y* dikenal sebagai variat kanonik. Kemudian kita dapat mendefinisikan operasi tingkat fitur kita sebagai penggabungan atau penjumlahan vektor fitur yang ditransformasikan, yaitu variat kanonik. CCA yang diturunkan dengan cara ini menawarkan dua sifat menarik:

(1) Matriks korelasi antar himpunan bersifat diagonal,
(2) Korelasi pada diagonal, yaitu korelasi antara X* dan Y*, dimaksimalkan.

Hal ini memastikan bahwa hubungan linier yang diperoleh antara X dan Y adalah optimal dan kita memiliki pemahaman tentang hubungan antara kedua himpunan variabel acak ini.

Fusi Akhir

Berbeda dengan fusi awal, pendekatan fusi akhir melatih pengklasifikasi terpisah untuk setiap saluran fitur yang ada dalam himpunan fitur kita $F = \{f_1, f_2, \dots, f_n\}$. Ini menghasilkan n modalitas pembelajaran dan pengklasifikasi yang berbeda. Setelah itu, pendekatan fusi akhir menggabungkan n skor klasifikasi yang berbeda ini menjadi skor klasifikasi akhir. Skema fusi akhir memiliki beberapa kekurangan yang jelas dibandingkan dengan skema fusi awal. Pertama-tama, fusi akhir menghasilkan jumlah pengklasifikasi yang lebih tinggi untuk dilatih yang meningkatkan beban komputasi. Kekurangan lainnya adalah bahwa pengklasifikasi tidak direpresentasikan dengan informasi dari modalitas atau fitur yang berbeda, sehingga korelasi antara modalitas tersebut tidak tercermin dalam skor keluaran pengklasifikasi. Terlepas dari kelemahan yang jelas ini, strategi fusi akhir dilaporkan memberikan hasil yang sebanding atau bahkan lebih baik daripada strategi fusi awal dalam beberapa aplikasi multimedia. Yang dkk. mengaitkan hal ini dengan sifat heterogen dan independen dari fitur multimedia.

Augmentasi Data

Augmentasi data adalah teknik yang dikembangkan untuk mencegah masalah overfitting selama pelatihan CNN. Selain itu, teknik ini juga banyak digunakan dalam fase pengujian. Terdapat berbagai teknik augmentasi; beberapa di antaranya adalah RGB jittering, penggunaan beberapa crop, evaluasi padat, rotasi dan pembalikan gambar. Pengaruhnya terhadap akurasi arsitektur masih kontroversial dan terbukti. Hampir semua penulis menyebutkan penggunaan teknik augmentasi yang intensif dalam studi mereka, sementara beberapa mengungkapkan informasi empiris yang kuat tentang marginalitas efek kinerjanya.

Dengan mengesampingkan situasi yang kontradiktif ini, kami lebih memilih untuk menghindari penggunaan augmentasi data dalam pekerjaan ini karena dua alasan lain. Pertama, augmentasi data, sebagian besar waktu, memerlukan menjalankan instance gambar apa pun melalui jaringan kami beberapa kali, yang tidak terlalu efisien, terutama ketika kita berurusan dengan big data. Kami percaya bahwa anggaran komputasi dapat digunakan dengan lebih baik untuk menjalankan beberapa CNN secara paralel sambil mencapai akurasi yang lebih tinggi dan peningkatan ketahanan. Kedua, penambahan data mencegah kita membandingkan hasil CNN dengan arsitektur dangkal seperti pengkodean bag-of-words menggunakan kuantisasi vektor atau pengkodean sparse, vektor Fisher atau pengkodean VLAD. Dalam upaya

sebelumnya dalam literatur klasifikasi gambar dan deteksi objek, penambahan data jarang digunakan.

8.4 EVALUASI KOMPARATIF EKSTRAKSI FITUR DAN FUSI PADA BENCHMARK GAMBAR

Kumpulan Data

Kami melaporkan hasil kami pada lima kumpulan data benchmark gambar yang berbeda. Kami menggunakan tantangan Pascal VOC, tantangan Pascal VOC 2012, Caltech101, Caltech256 dan Caltech UCSD Birds kumpulan data benchmark gambar. Kumpulan data Caltech101 berisi 101 kategori objek yang berbeda dan satu latar belakang yang kami gunakan sebagai kategori normal dalam eksperimen kami. Caltech101 berisi total 9.144 gambar dan memiliki 31 hingga 800 gambar per kategori. Selain itu, sebagian besar kategori memiliki lebih dari 50 gambar. Resolusi gambar rata-rata sekitar 300 × 200 piksel dan setiap gambar hanya berisi satu objek. Caltech101 adalah salah satu dataset yang paling banyak diadaptasi oleh komunitas computer vision, untuk tugas pengenalan dan klasifikasi objek. Caltech101 tidak menyediakan daftar gambar untuk pembagian kasus pelatihan dan pengujian.

Praktik umum yang digunakan dalam literatur adalah membagi dataset secara acak menjadi dua dan kemudian menggunakan satu untuk pelatihan dan yang lainnya untuk pengujian. Untuk mencegah efek favoritisme dari pembagian acak, hasil dilaporkan pada setidaknya tiga pembagian yang berbeda. Validasi silang sama sekali tidak digunakan. Rezim pembagian yang paling sering digunakan adalah menggunakan 30 sampel pelatihan dan sisanya untuk pengujian. Karena hasil disajikan menggunakan akurasi rata-rata per kelas, jumlah gambar yang tidak merata yang dipilih dalam fase pengujian dihilangkan.

Dataset Caltech256 merupakan perluasan dari Caltech101 dan berisi 256 kategori objek yang mencakup total 30.607 gambar. Caltech256 memiliki struktur yang sama dengan Caltech101 tetapi dengan distribusi gambar yang lebih baik. Dibandingkan dengan Caltech101, jumlah minimum gambar dalam setiap kategori ditingkatkan dari 31 menjadi 80. Prosedur validasi hampir sama dengan Caltech101, satu-satunya perbedaan adalah pilihan untuk pembagian data pelatihan adalah menggunakan 60 gambar pelatihan dan sisanya untuk pengujian.

Birds-200 adalah salah satu dataset paling populer yang digunakan untuk klasifikasi gambar detail dalam literatur. Dataset ini terdiri dari 200 spesies burung yang berbeda dan berisi 11.788 gambar. Dataset Pascal VOC 2007 adalah dataset benchmark lain untuk klasifikasi dan lokalisasi gambar. Pascal VOC 2007 berisi 20 kelas dan total 9.963 gambar dan 24.640 objek yang dianotasi. Ini adalah dataset yang lebih sulit jika dibandingkan dengan variasi Caltech meskipun memiliki kelas objek yang jauh lebih sedikit. Hal ini disebabkan oleh kekacauan dan oklusi dalam gambar serta rentang spasial objek yang rendah dalam gambar pelatihan. Perbedaan lain dari dataset Caltech adalah bahwa Pascal VOC menyediakan daftar pelatihan sehingga semua orang menggunakan data pelatihan dan pengujian yang sama untuk menyajikan hasil mereka. Pascal VOC juga menggunakan ukuran presisi rata-rata alih-alih ukuran akurasi sederhana yang digunakan dalam dataset Caltech. Pascal VOC 2012 adalah versi

terbaru dari Pascal VOC 2007 dan berisi lebih banyak gambar. Kami menyajikan hasil untuk kedua dataset demi perbandingan.

Evaluasi Fitur yang Diekstrak

Pada bagian ini, kami membandingkan berbagai metode ekstraksi fitur. Kami mengekstrak fitur dari lapisan terhubung penuh fc6 dan fc7 dari VGG19. Kami juga mengekstrak fitur dari semua lapisan konvolusional dari arsitektur yang sama. Selain melaporkan angka untuk semua lapisan ini, kami juga menggabungkan semua fitur menjadi satu modalitas menggunakan penggabungan fitur dan melaporkan hasil dari mekanisme fusi sederhana ini bersama dengan lapisan lainnya. Menurut hasil yang ditunjukkan pada Tabel 8.3, skema ekstraksi fitur sum-pooling dari lapisan konvolusional mencapai kinerja yang cukup baik mengingat kesederhanaannya. Kami menerapkan sum-pooling pada peta fitur yang diekstrak dari lapisan konvolusional arsitektur CNN untuk mengekstrak fitur gambar tunggal yang mewakili gambar dan menggunakan SVM linier untuk klasifikasi fitur yang diekstrak.

Menggabungkan informasi dari hasil ekstraksi fitur dari lapisan konvolusional dengan lapisan yang terhubung sepenuhnya juga bermanfaat untuk meningkatkan kinerja klasifikasi akhir. Hal ini lebih jelas terlihat pada beberapa dataset yang tidak memiliki banyak kemiripan dengan dataset pra-pelatihan. Lebih spesifiknya, arsitektur VGG19 kami dilatih pada dataset ImageNet yang berisi banyak contoh yang ada di Pascal 2007, sehingga kombinasi fitur tidak terlalu membantu untuk Pascal 2007. Di sisi lain, dataset Caltech memperoleh tingkat pengenalan yang lebih tinggi saat menggunakan kombinasi fitur karena hampir tidak termasuk dalam dataset ImageNet. Kami juga telah mengevaluasi max-pooling dalam kasus sum-pooling dan kami menyimpulkan bahwa max-pooling lebih rendah daripada sum-pooling. Hasil kami pada Pascal 2007 menggunakan arsitektur AlexNet dapat dilihat pada Tabel 8.4.

Tabel 8.3 Kinerja klasifikasi beberapa lapisan arsitektur VGG19.

Nama Lapisan	Pascal VOC 2007	Pascal VOC 2012	Caltech UCSD Birds	Caltech101
conv1	0.31	0.28	0.15	0.62
conv2	0.50	0.47	0.27	0.66
conv3	0.62	0.58	0.36	0.67
conv4	0.70	0.68	0.44	0.70
conv5	0.77	0.73	0.50	0.71
fc6	0.81	0.79	0.57	0.91
fc7	0.81	0.78	0.55	0.89
all conv. layers + fc6	0.84	0.81	0.60	0.90

Tabel 8.4 Hasil penggabungan fitur lapisan konvolusi Alex-Net pada Caltech101.

Nama Layer	Sum Pooling	Max Pooling
conv1	0.33	0.25
conv2	0.52	0.54
conv3	0.64	0.70
conv4	0.69	0.75
conv5	0.73	0.74

Perbandingan Kinerja Arsitektur CNN

Kami mengevaluasi kinerja arsitektur AlexNet, VGG19, GoogleNet, dan ResNet152. Tabel 8.5 merangkum temuan kami. Kami menyajikan hasil klasifikasi menggunakan dua jenis fitur yang berbeda. Kumpulan hasil pertama diperoleh menggunakan lapisan fully-connected pertama dari setiap arsitektur. Kumpulan hasil kedua menunjukkan skema kombinasi fitur kami yang menggunakan lapisan konvolusional selain lapisan fully-connected pertama yang hasilnya telah kami sajikan di bagian sebelumnya.

Tabel 8.5 Kinerja klasifikasi berbagai arsitektur CNN. Tanda "*" pada arsitektur CNN mewakili fitur yang diekstrak dari beberapa lapisan konvolusional dan fully-connected, sedangkan yang lainnya mewakili fitur yang diekstrak hanya dari lapisan fully-connected pertama.

CNN Architecture	Pascal VOC 2007	Pascal VOC 2012	Caltech UCSD Birds	Caltech101	Caltech256
AlexNet	0.70	0.68	0.48	0.89	0.72
VGG19	0.81	0.79	0.57	0.91	0.80
GoogleNet	0.80	0.78	0.55	0.90	0.80
ResNet152	0.82	0.80	0.59	0.91	0.83
AlexNet*	0.73	0.71	0.49	0.93	0.72
VGG19*	0.84	0.81	0.60	0.90	0.81
GoogleNet*	0.84	0.79	0.59	0.91	0.82
ResNet152*	0.85	0.81	0.61	0.93	0.85

Mengevaluasi Skema Fusi

Setelah mengevaluasi skema ekstraksi fitur, kita beralih ke masalah fusi fitur. Pertama, kita mengekstrak fitur dari dua arsitektur CNN, yaitu AlexNet dan VGG19. Kita membandingkan hasil dari tiga operasi fusi sederhana yang dirangkum dalam Tabel 8.2 dan fusi fitur berbasis CCA yang telah kita jelaskan sebelumnya. Kita merangkum hasil kita dalam Tabel 8.6. Metode fusi fitur sederhana untuk fitur lapisan CNN secara mengejutkan berkinerja baik jika dibandingkan dengan strategi fusi yang lebih kompleks dari fusi berbasis CCA. Ini jelas merupakan hasil dari kekuatan representasi yang kuat dari fitur lapisan CNN. Kita juga menunjukkan bahwa fusi awal secara konsisten mengungguli fusi akhir dalam semua evaluasi dan dataset. Ini sebagian besar disebabkan oleh kardinalitas rendah dari fitur lapisan CNN yang diekstrak di mana pengklasifikasi linier mampu membedakan secara efektif.

Tabel 8.6 Kinerja fusi awal dan akhir dari arsitektur CNN yang digunakan pada dataset terpilih. Arsitektur CNN yang ditandai dengan "*" menggambarkan hasil dari Tabel 8.5, sedangkan sisanya mewakili fusi dari kedua arsitektur tersebut.

CNN Architecture / Method	Pascal VOC 2007	Pascal VOC 2012	Caltech UCSD Birds	Caltech 101	Caltech 256
AlexNet*	0.73	0.71	0.49	0.93	0.72
VGG19*	0.84	0.81	0.60	0.90	0.81

Sum Fusion	0.83	0.80	0.62	0.93	0.82
Max Fusion	0.81	0.80	0.61	0.92	0.81
Concatenate Fusion	0.84	0.81	0.63	0.93	0.83
CCA Fusion	0.82	0.79	0.61	0.91	0.82
Late Fusion	0.82	0.79	0.61	0.92	0.82

Tabel 8.7 Kinerja waktu eksekusi arsitektur CNN yang digunakan.

Jaringan	Skala	(FPS)
AlexNet	227 x 227	210
GoogleNet	224 x 224	58
VGG19	224 x 224	47
ResNet152	227 x 227	11

Kompleksitas Komputasi

Mengingat anggaran komputasi selalu merupakan sumber daya yang langka, sumber daya komputasi harus digunakan dengan bijak bahkan ketika perhatian utama adalah untuk meningkatkan kualitas. Untuk memberikan sedikit pencerahan pada sisi gelap inferensi CNN, kami mengumpulkan beberapa metrik waktu eksekusi selama pengujian kami. Pengujian dilakukan pada GPU GTX980 yang terpasang pada CPU i5-4460 kelas bawah dengan 4 inti yang berjalan pada kecepatan clock 3.20 GHZ.

Peringkat berasal dari pengujian ujung-ke-ujung dari setiap model CNN, yaitu dari gambar yang berjalan melalui semua lapisan arsitektur yang sesuai. Fitur diekstrak dari lapisan terhubung penuh terakhir dari semua arsitektur. Pada Tabel 8.7, kami membagikan peringkat absolut "frame per detik" dari setiap arsitektur CNN. Hasil menunjukkan bahwa kompleksitas komputasi dari arsitektur yang digunakan sebanding dengan skala arsitektur seperti yang diharapkan.

8.5 SISTEM PENGENALAN KONSEP VISUAL BERBASIS FUSI FITUR PADA ARSITEKTUR CNN

Jaringan konvolusional telah menunjukkan potensi besar dalam berbagai tugas pemahaman gambar akhir-akhir ini, dan dalam artikel ini kami memberikan informasi empiris yang luas tentang kinerja empat desain CNN yang berbeda, yaitu AlexNet, VGG, GoogleNet, dan ResNet. Kami mengungkapkan hasil klasifikasi menggunakan model CNN ini pada berbagai dataset. Kontribusi lain dari pembelajaran kami adalah kami mengusulkan sistem pengenalan konsep visual lengkap berdasarkan arsitektur CNN yang menggunakan skema fusi awal untuk meningkatkan akurasi pengenalan. Secara khusus, kami meningkatkan hasil model CNN tunggal dengan menggabungkan informasi dari lapisan yang berbeda dari model CNN yang diberikan. Kami mengamati bahwa fitur yang dipelajari oleh arsitektur CNN yang berbeda saling melengkapi dan fusi yang tepat dapat menghasilkan peningkatan kinerja. Kami mengungkapkan bahwa operasi penjumlahan dan max pooling sederhana mampu mengekstrak informasi relevan yang berada di lapisan jaringan yang berbeda tanpa mengurangi akurasi dan dapat bersaing dengan pengkodean tingkat yang lebih tinggi yang menghasilkan jumlah kardinalitas data yang relatif lebih tinggi. Dengan menggunakan skema

fusi yang kami usulkan, dimungkinkan untuk mencapai hasil terbaik tanpa menggunakan teknik augmentasi data yang ekstensif. Kami akan menyelidiki penggabungan fitur dengan pengkodean yang lebih canggih dan memberikan hasil empiris yang luas yang membandingkan skema penggabungan sederhana dan lanjutan. Selain itu, kami berencana untuk memperluas solusi yang kami usulkan ke domain video di mana penggabungan data multimodal tetap menjadi tantangan teknik yang penting.

BAB 9
JARINGAN NEURAL DALAM UNTUK SOLUSI INDUSTRI REAL-TIME

Tujuan artikel ini adalah untuk menyelidiki penerapan algoritma evolusioner pada desain pengontrol industri real-time. Pembelajaran mendalam (*deep learning*/DL) saat ini telah mapan sebagai alat yang berguna untuk mengatasi banyak masalah praktis. Hal ini telah mendorong pengembangan metode pencarian arsitektur neural (*neural architecture search*/NAS) untuk mengotomatisasi aktivitas pencarian model. CATNeuro adalah algoritma NAS yang didasarkan pada konsep evolusi graf yang dirancang oleh *Neuroevolution of Augmenting Topologies* (NEAT) tetapi didorong oleh algoritma budaya (*cultural algorithm*/CA) sebagai penggerak evolusi. CA adalah kerangka kerja optimasi stokastik berbasis jaringan yang terinspirasi oleh pemecahan masalah dalam budaya manusia.

Distribusi pengetahuan (*knowledge distribution*/KD) di seluruh jaringan model graf merupakan kunci keberhasilan pemecahan masalah dalam sistem CAT. Dua mekanisme alternatif untuk KD di seluruh jaringan digunakan. Satu mendukung kerja sama (CATNeuro) dalam jaringan dan yang lainnya persaingan (*competition*/WM). Untuk menguji kelayakan setiap konfigurasi sebelum digunakan dalam pengaturan industri, keduanya diterapkan pada desain pengontrol waktu nyata untuk permainan pertarungan dua dimensi.

Meskipun keduanya mampu mengalahkan program AI yang disertakan dengan permainan pertarungan tersebut, metode kooperatif secara statistik berkinerja lebih baik. Akibatnya, metode ini digunakan untuk melacak pergerakan trailer. (dalam arah lateral dan vertikal) menggunakan kamera yang dipasang pada kendaraan traktor yang menarik trailer. Dalam aplikasi waktu nyata kedua ini (gerakan trailer), konfigurasi CATNeuro dibandingkan dengan metode evolusi NEAT (elitist) asli. CATNeuro ditemukan berkinerja secara statistik lebih baik daripada NEAT dalam banyak aspek desain termasuk kerugian pelatihan model, ukuran parameter model, dan konsistensi struktur model secara keseluruhan. Dalam kedua skenario, peningkatan kinerja dikaitkan dengan peningkatan keragaman model karena interaksi sumber pengetahuan CA baik secara kooperatif maupun kompetitif.

9.1 PEMBELAJARAN MENDALAM DAN NAS

Fokus artikel ini adalah untuk menyelidiki penerapan algoritma evolusi pada desain pengendali industri waktu nyata. “Pembelajaran mendalam” (*deep learning*/DL) saat ini telah mapan sebagai alat yang berguna untuk mengatasi banyak masalah praktis. Hal ini telah mendorong pengembangan metode pencarian arsitektur neural (*neural architecture search*/NAS) untuk mengotomatiskan aktivitas pencarian model.

Tugas desain representatif yang digunakan di sini melibatkan model proses kontinu, yaitu pergerakan trailer yang terpasang pada kendaraan penumpang. Menurut Administrasi Keselamatan Nasional AS, lebih dari 400 orang meninggal setiap tahun akibat kecelakaan tersebut dan lebih dari 20.000 orang mencari perawatan medis sebagai akibatnya. Kecelakaan ini merugikan perekonomian lebih dari Rp 20 triliun setiap tahunnya. Ada banyak penyebab

berbeda dari kecelakaan ini, seperti kelebihan muatan, pemasangan yang tidak tepat, beban yang tidak seimbang, pusat gravitasi yang tinggi, dan kesalahan perhitungan belokan dan jarak pengereman karena kesalahan pengemudi.

Meskipun teknologi mengemudi otonom telah berfokus pada keselamatan mobil penumpang, ini jelas merupakan perluasan penting dan tidak sepele dari pekerjaan ini. Meskipun sudah mapan, DL masih membutuhkan banyak upaya dan waktu manusia untuk menemukan hyperparameter model yang tepat untuk masalah yang kompleks. Dari semua hyperparameter, topologi model adalah Kontributor paling signifikan terhadap kinerja. Secara umum, input dan output (terminal) jaringan diberikan tetapi topologi internal jaringan masih dapat dikembangkan. Desain hyperparameter untuk model DL dapat dilihat dari tiga perspektif berbeda:

(i) Pertama, pada tingkat arsitektur, topologi jaringan dapat dilihat sebagai grafik berarah, di mana node internal adalah "lapisan" seperti LSTM atau drop-off, "sel" seperti pada ResNet, BERT, dan mungkin "modul", misalnya perhatian.

(ii) Pada tingkat node dalam grafik, terdapat parameter untuk setiap node, misalnya jumlah neuron lapisan, skema inisialisasi, skema aktivasi, jika ada, dll.

(iii) Pada tingkat pembelajaran, terdapat parameter tingkat pelatihan yang menentukan bagaimana data diproses melalui jaringan node, misalnya laju pembelajaran, ukuran minibatch, momentum, dll.

Hampir tanpa kecuali, penyetelan hyperparameter pada setiap level ini diperlukan untuk setiap model DL yang signifikan. Dengan demikian, sudah ada beberapa kerangka kerja NAS dari akademisi dan industri (misalnya Google dan Microsoft). Di antara berbagai pendekatan, neuroevolusi merupakan pendekatan yang menarik untuk DL karena struktur model yang efektif atau optimal "ditemukan" melalui evolusi dan kompleksitas progresif dari grafik terarah.

Tidak seperti beberapa pendekatan NAS berbasis DL (misalnya DARTS), neuroevolusi hanya memberikan sedikit batasan a priori pada struktur model. Dengan demikian, pemodelan neuroevolusi dapat menemukan struktur baru dan lebih efisien untuk menyelesaikan masalah yang ada meskipun dengan biaya komputasi yang lebih besar karena ruang pencarian yang diperluas. Miikkulainen telah menunjukkan bahwa pemodelan neuroevolusi kompetitif dengan model DL buatan tangan. Awalnya, strategi sederhana neuroevolusi efektif karena memaksimalkan jumlah percobaan dengan anggaran komputasi yang ada. Baik struktur model maupun bobot parameternya dipelajari/dievaluasi secara bersamaan di setiap langkah.

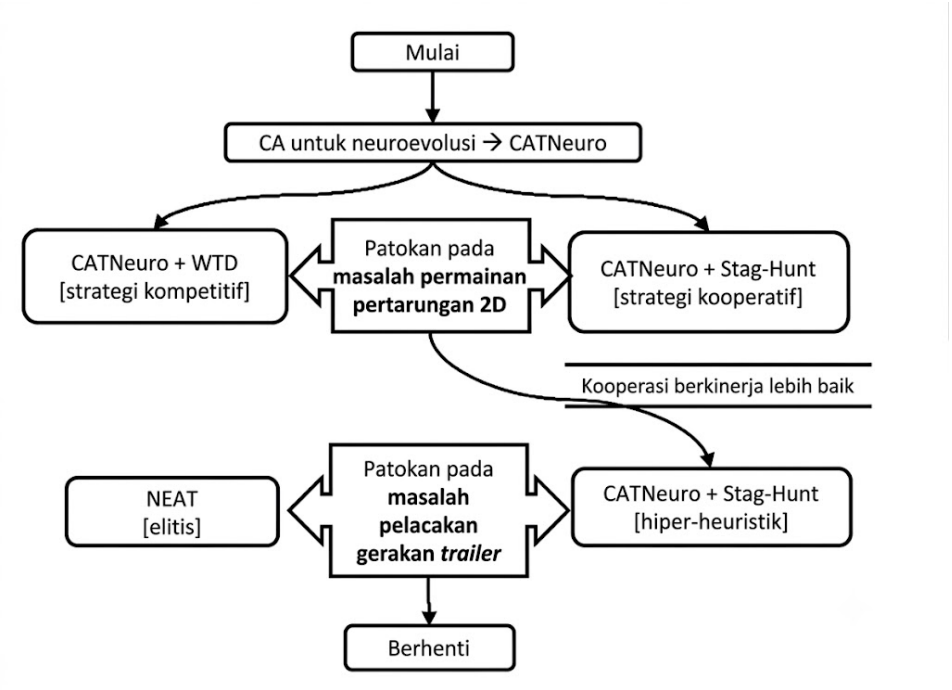
Sebaliknya, NAS kontemporer adalah proses pencarian dua langkah. Pertama, struktur model diusulkan. Selanjutnya, bobot parameter optimal dipelajari dengan melatih model tersebut secara offline. Waktu pencarian didominasi oleh pelatihan model. Dengan pengaturan ini, Sumber daya komputasi dapat dialihkan secara efektif ke strategi evolusi yang sesuai untuk mengusulkan struktur model yang lebih informatif tanpa secara signifikan memengaruhi waktu pencarian secara keseluruhan.

Kerangka kerja yang digunakan di sini adalah untuk mengarahkan proses pencarian NAS kontemporer menggunakan algoritma evolusi yang dimotivasi secara sosial, yaitu algoritma budaya (CA). CA adalah kerangka kerja optimasi stokastik yang terinspirasi oleh proses budaya

manusia dan hewan untuk bertahan hidup dan beradaptasi. Algoritma ini telah diterapkan pada beragam domain aplikasi waktu nyata, misalnya optimasi numerik, arkeologi, biologi, pengenalan gerak tubuh, dan visi komputer, untuk menyebutkan beberapa saja. CA sangat cocok untuk aplikasi seperti NAS di mana waktu pelatihan untuk suatu model jauh melebihi waktu untuk mencari model baru. Hal ini karena pengetahuan yang dikumpulkan sepanjang proses dalam komponen budaya dapat digunakan untuk secara sistematis menyuntikkan entropi ke dalam populasi individu yang mendukung budaya tersebut. Ini adalah karakteristik dari desain banyak sistem otonom yang kompleks.

Dalam karya ini, disajikan versi CA yang secara khusus disesuaikan dengan proses pencarian NAS, yaitu CATNeuro. CATNeuro dirancang untuk mengembangkan grafik terarah untuk model DL. CATNeuro berisi beberapa ekstensi dari kerangka kerja *Neuroevolution of Augmenting Topologies* (NEAT) secara khusus untuk mendukung evolusi model DL. Diagram skematis dari strategi pembelajaran keseluruhan untuk karya ini disajikan pada Gambar 9.1.

Mengikuti diagram pada Gambar 9.1, artikel ini disusun seperti yang dijelaskan selanjutnya. Karena CATNeuro memperluas fungsionalitas dasar NEAT, operasi grafik di bawah NEAT menyajikan versi CATNeuro dari CA yang memperluas fungsionalitas NEAT. Karena CA termotivasi secara sosial dan digerakkan oleh jaringan, salah satu keputusan desain utama adalah bagaimana mendistribusikan pengetahuan di antara agen dalam populasi. Di sini, CATNeuro diimplementasikan menggunakan dua mekanisme distribusi pengetahuan (KD) yang berbeda. Pendekatan Majority Win atau *Wisdom of the Crowds* (WTD) mendukung persaingan antar sumber pengetahuan (KS). Pendekatan lainnya, Stag-Hunt, mendukung kerja sama dan persaingan antar KS. KD Mekanisme (Stag-Hunt dan WTD) dibahas di Bagian 9.4. Sebelum penerapan CATNeuro pada masalah desain industri, kedua pendekatan tersebut dibandingkan relatif terhadap desain pengontrol waktu nyata untuk permainan pertarungan dua dimensi (2D), FightingICE. Dengan lebih dari 56 aksi berbeda yang harus dikendalikan terhadap pemain mana pun, hal ini dianggap sebagai alat benchmarking yang sesuai. Bagian 9.5 menjelaskan paradigma pembelajaran penguatan yang digunakan untuk melatih sistem beserta hasilnya.



Gambar 9.1 Diagram alur proses pembelajaran.

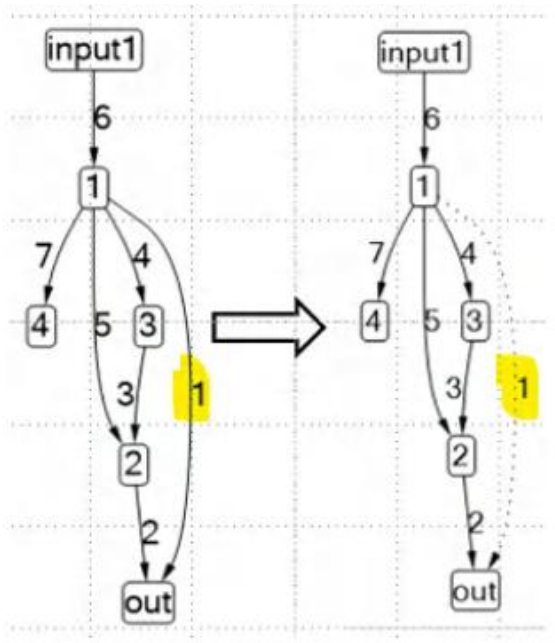
Seperti yang ditunjukkan pada Gambar 9.10, distribusi Stag-Hunt secara statistik mengungguli pendekatan WTD. Oleh karena itu, distribusi ini dipilih sebagai platform untuk mendesain pengontrol pelacakan gerak. Bagian 6 dan subbagiannya membahas masalah pelacakan gerak trailer, pembuatan data pelatihan, pengaturan eksperimen, dan hasil eksperimen. Masalah pelacakan gerak trailer digunakan sebagai dasar perbandingan antara CA (dikonfigurasi dengan Stag-Hunt) dan metode evolusi graf NEAT (elitist) asli. Bagian 9.7 merangkum artikel ini dan menyarankan arah pembelajaran di masa mendatang.

9.2 OPERASI GRAF DI BAWAH NEAT

Operasi graf CATNeuro sama dengan operasi graf untuk *Neuroevolution of Augmenting Topologies* atau NEAT. Operasi ini hanya dijelaskan secara singkat dan skematis dalam artikel ini. Pembaca dapat merujuk ke artikel asli oleh penulis NEAT untuk detail operasi. Ide inti NEAT adalah mengembangkan grafik melalui "kompleksifikasi" progresif yaitu, simpul dan koneksi ditambahkan tetapi tidak dihapus. Operasi grafik CATNeuro/NEAT adalah sebagai berikut:

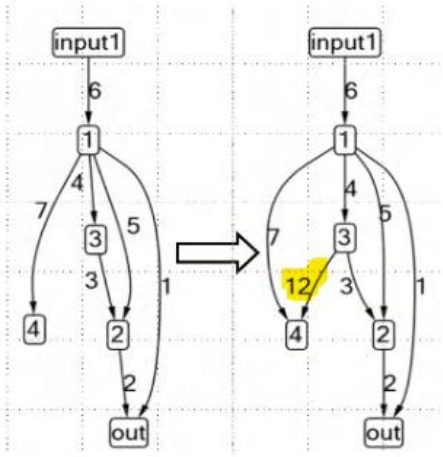
- Mengaktifkan/menonaktifkan koneksi
- Menambahkan koneksi
- Menambahkan simpul
- Crossover
- Mengubah parameter.

Dalam operasi mengaktifkan/menonaktifkan koneksi (Gambar 9.2), koneksi antara dua simpul dapat diaktifkan atau dinonaktifkan. Koneksi "nonaktif" dianggap tidak ada untuk penerjemahan ke model konkret. Dalam operasi menambahkan koneksi, dua simpul yang sebelumnya tidak terhubung dihubungkan ke koneksi baru (Gambar 9.3).

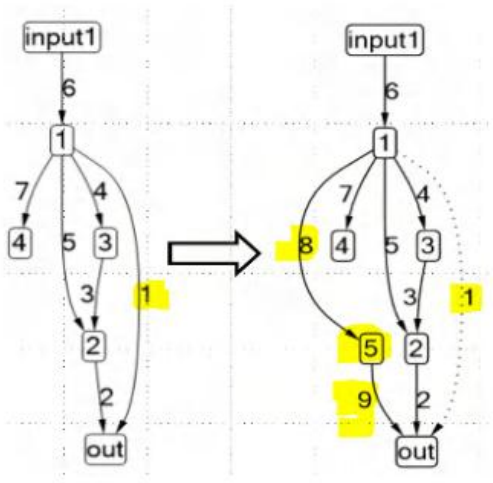


Gambar 9.2 Alihkan koneksi #1 (sebelum kiri, setelah kanan).

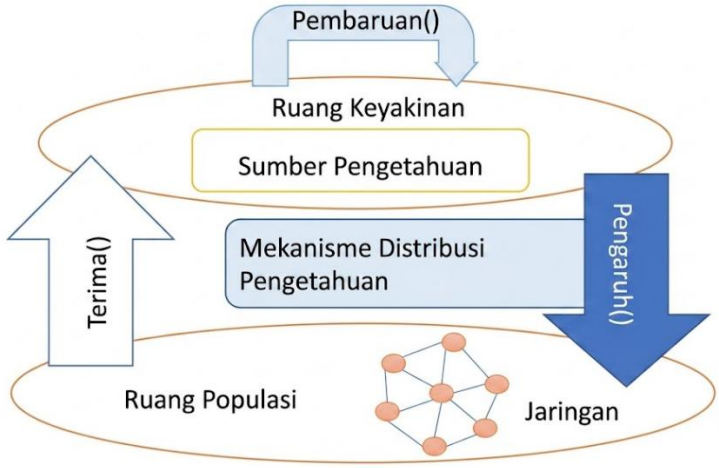
Dalam operasi menambahkan simpul, koneksi yang dipilih secara konseptual dipisahkan dengan menambahkan simpul baru di tengahnya. Hal ini dilakukan dengan mengubah koneksi yang dipilih menjadi "mati" dan menambahkan dua koneksi baru dan satu node baru seperti yang ditunjukkan pada Gambar 9.4. Di bawah NEAT, setiap koneksi baru diberi pengenal unik yang meningkat secara monoton yang disebut nomor inovasi. Dalam operasi crossover (tidak ditunjukkan), dua grafik digabungkan dengan menyelaraskan nomor inovasi grafik dalam semacam pengurutan topologi. Hanya koneksi maju yang diizinkan, yaitu koneksi baru tidak boleh membuat siklus. Dalam parameter mutate, struktur grafik tidak dimodifikasi, melainkan hyperparameter global (misalnya laju pembelajaran) dan parameter internal node grafik dievolusikan. Untuk DL, parameter node dapat berupa dimensi lapisan (jumlah neuron), jenis aktivasi, dll.



Gambar 9.3 Tambahkan koneksi #12 dari node 3 ke node 4 (sebelum kiri, setelah kanan).



Gambar 9.4 Tambahkan node — koneksi #1 "diganti" dengan koneksi #8 dan #9 yang menghubungkan node baru 5 ke sumber dan target dari #1.



Gambar 9.5 Arsitektur CA.

9.3 CA UNTUK EVOLUSI GRAFIK

Bagian ini menjelaskan adaptasi konsep dan ide CA (*cultural algorithm)* ke domain NAS (*neural architecture search)*. Ciri khas dari CA adalah (a) ruang populasi yang terhubung secara "sosial" dan (b) jaringan Ruang Kepercayaan yang merupakan repositori pengetahuan. Gambar 9.5 menunjukkan arsitektur CA di mana panah mewakili fungsi. Bagaimana komponen dasar CA diadaptasi untuk mendukung DL di sini akan dibahas sekarang:

- Ruang Kepercayaan: Kumpulan KS.
- Sumber Pengetahuan: Metaheuristik atau strategi pencarian yang menggerakkan individu populasi terkait melalui ruang pencarian dengan cara yang unik. Setiap KS mempertahankan informasinya sendiri untuk menggerakkan proses pencarian. Secara umum, ada lima jenis KS: (a) Situasional, (b) Domain, (c) Sejarah, (d) Normatif, dan (e) Topografi.

- Ruang Populasi: Kumpulan agen individu, yang terhubung dalam konfigurasi tertentu. Dalam CATNeuro, setiap agen individu berisi grafik terarah dan dikaitkan dengan KS pada setiap langkah waktu. KS menggerakkan individu melalui ruang pencarian dengan mengembangkan grafik terkaitnya. Agen dapat secara langsung memengaruhi tetangga yang terhubung langsung dengannya di jaringan.
- Fungsi Terima: Dalam setiap siklus generasi, fungsi Terima memilih individu yang paling menarik dari populasi ke dalam Ruang Kepercayaan. Di sini, kerugian pelatihan jaringan saraf digunakan sebagai ukuran utama kebugaran. Namun, sistem ini juga memungkinkan penyertaan ukuran model saat memberi peringkat individu untuk mendukung model yang hemat.
- Fungsi Perbarui: Perbarui mengekstrak informasi dari individu yang diterima dan memperbarui KS. Setiap KS mempertahankan kumpulan informasi yang berbeda dan karenanya pembaruan bersifat spesifik untuk setiap KS.
- Fungsi Pengaruh: Pengaruh pertama-tama mendistribusikan pengetahuan di ruang populasi melalui mekanisme KD yang dikonfigurasi, dan kemudian mengembangkan individu melalui KS terkaitnya untuk generasi berikutnya.
- Mekanisme KD: Mekanisme KD mengaitkan KS dengan setiap individu populasi. Ini adalah heuristik yang dapat dipasang yang dapat dan biasanya memanfaatkan jaringan populasi.

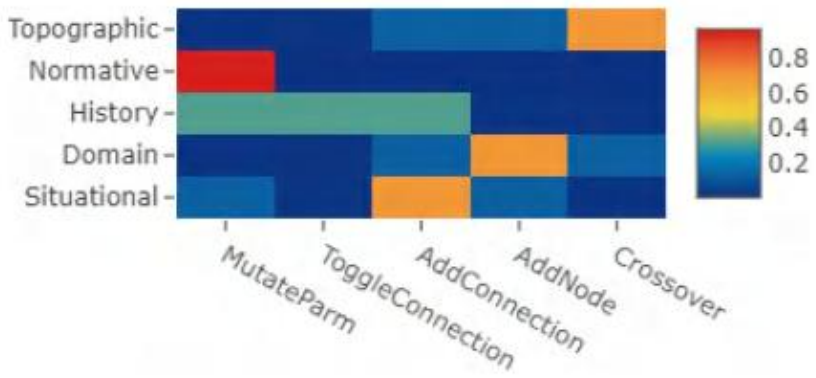
KS adalah strategi pencarian metaheuristik, yang terinspirasi oleh proses dalam budaya manusia dan hewan. Setiap KS memodifikasi individu terkait dengan cara spesifiknya sendiri. Secara umum, ada lima jenis KS: (a) Situasional, (b) Domain, (c) Sejarah, (d) Normatif, dan (e) Topografis. KS Situasional melacak individu "teladan" yang perilakunya dapat ditiru pada generasi mendatang. KS Domain mewakili pengetahuan dan batasan khusus yang mengarahkan tindakan. KS Sejarah menangkap memori episodik yang mencerminkan siklus dalam proses pemecahan masalah. KS Normatif mewakili pola representatif perilaku agen. KS Topografi adalah deskripsi tingkat tinggi dari lanskap pencarian.

KS-KS ini bekerja bersama untuk memengaruhi arah pencarian. Dalam CATNeuro, fungsi pengaruh adalah penerapan satu atau lebih operasi grafik pada individu saat ini untuk menciptakan agen individu baru untuk generasi berikutnya. Ini membutuhkan pemetaan dari setiap KS ke operasi grafik yang ditentukan sebelumnya. Untuk CATNeuro, pemetaan ini bersifat stokastik, yaitu distribusi probabilitas atas operasi grafik dikaitkan dengan setiap KS seperti yang diilustrasikan pada Gambar 9.6. Setiap baris dalam gambar sesuai dengan probabilitas penggunaan dengan masing-masing dari lima operasi grafik untuk setiap KS. Dalam contoh ini, kemungkinan bahwa operator tertentu akan dipilih ditunjukkan oleh kode warna. Pada saat pengaruh, KS mengambil sampel dari distribusi ini dan menerapkan operasi grafik yang dipilih. Kode warna mewakili frekuensi relatif penggunaan masing-masing KS secara berurutan dalam eksperimen.

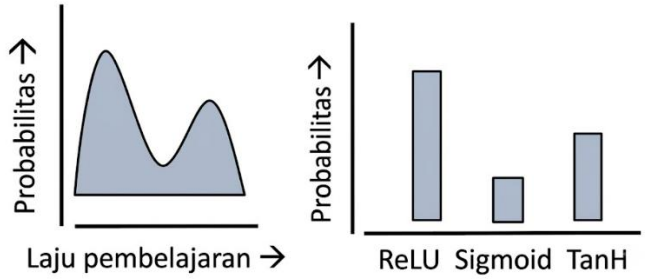
Gambar tersebut mengilustrasikan keseimbangan antara eksplorasi dan eksploitasi yang dibutuhkan oleh pengoptimal stokastik untuk secara efektif mencari ruang masalah kompleks seperti ini. Setiap KS beroperasi di lokasi yang berbeda di sepanjang kontinum

eksplorasi-eksploitasi. KS CAT-Neuro diberi peringkat berdasarkan bias eksploratif. Memang, mekanisme distribusi KD Stag-Hunt membutuhkan peringkat eksplisit seperti itu agar dapat beroperasi secara efektif. Operasi grafik juga dapat diberi peringkat berdasarkan eksplorasi dan eksploitasi.

Operasi yang mengubah struktur grafik, misalnya menambahkan node dan crossover, harus bersifat eksploratif (secara agregat) dibandingkan dengan operator yang tidak membuat perubahan grafik (memutasi parameter) atau perubahan struktural yang relatif kecil, misalnya menambahkan koneksi dan mengalihkan koneksi. Pemetaan pada Gambar 9.6 mencerminkan bias eksploratif dari setiap KS. Misalnya, KS Topografi adalah yang paling eksploratif. Ia memberikan bobot terbesar pada operator crossover, yang merupakan operasi yang paling mengganggu. KS Situasional dan KS Sejarah relatif eksploitatif dan, oleh karena itu, distribusi operasi grafnya bias terhadap operasi yang terkait dengan koneksi, yang membuat perubahan yang relatif kecil pada graf. KS Normatif adalah yang paling eksploitatif dan sangat terkait dengan parameter mutate yang tidak mengubah struktur graf seperti yang ditunjukkan pada Gambar 9.6. Bobot pemetaan dihasilkan dengan pengujian empiris yang terbatas. Kemungkinan ada ruang untuk perbaikan — sebuah subjek pembelajaran di masa mendatang.



Gambar 9.6 Pemetaan KS ke operasi grafik yang ditampilkan sebagai peta panas — setiap baris adalah distribusi probabilitas atas operasi grafik.



Gambar 9.7 Contoh distribusi probabilitas bernilai kontinu dan diskrit untuk pengambilan sampel berbasis kepadatan.

Nomor inovasi awalnya berasal dari neuroevolusi. Setiap kali koneksi ditambahkan (untuk mengembangkan grafik), koneksi tersebut diberi ID unik yang meningkat secara

monoton. Ini memungkinkan seseorang untuk melacak evolusi suatu grafik. Di antara kegunaan lainnya, nomor inovasi digunakan untuk membandingkan dua grafik dan menyelaraskannya untuk operasi crossover (yaitu menggabungkan dua grafik). Sebuah koneksi tidak dihapus sehingga nomor inovasi tidak berubah (meskipun koneksi dapat dimatikan).

Dalam CATNeuro, nomor inovasi juga digunakan sebagai jangkar untuk mengumpulkan statistik tentang kinerja yang terkait dengan parameter node target. Salah satu parameternya bisa berupa jumlah neuron. Jika grafik dalam populasi dengan nomor inovasi tertentu berkinerja lebih baik dengan misalnya 10 neuron daripada grafik dengan nilai lain untuk parameter yang sama, maka 10 akan memiliki peluang lebih tinggi untuk dipilih ketika node target dimutasi selanjutnya.

Pengambilan sampel kepadatan kernel hanyalah cara untuk menggunakan statistik yang disebutkan di atas untuk pengambilan sampel. Misalkan kita telah mencoba tiga nilai untuk suatu parameter dan mencatat kebugaran rata-rata individu yang terkait dengan setiap nilai, seperti yang ditunjukkan di bawah ini.

Contoh distribusi jumlah neuron untuk pengambilan sampel kepadatan kernel.	
Jumlah neuron	Rata-rata kebugaran individu terkait
10	1.2
12	2.0
15	2.1

Kita dapat menggunakan ini untuk membuat distribusi probabilitas dengan menggunakan nilai fitness sebagai bobot. Kita juga mengasumsikan bahwa nilai-nilai tersebut memiliki distribusi Gaussian. Ini memberi kita kurva distribusi probabilitas kontinu. Kemudian kita dapat mengambil sampel darinya (seperti roda putar tetapi kontinu). Nilai fitness yang lebih tinggi akan memiliki "massa" yang lebih tinggi, oleh karena itu, nilai yang diambil sampelnya kemungkinan akan mendekati nilai tersebut. Dalam contoh di atas, nilai yang mendekati 12–15 neuron akan memiliki kemungkinan lebih tinggi untuk dipilih untuk mutasi berikutnya.

Karena sifat proses evolusi grafik NEAT, angka inovasi yang terkait dengan setiap koneksi dibagi di banyak grafik dalam jaringan populasi dan dapat dianggap sebagai jangkar semantik. Data untuk angka inovasi yang sama di banyak grafik yang berisi inovasi tersebut dikumpulkan untuk peta kepadatan. Peta-peta ini diambil sampelnya untuk nilai parameter yang menjanjikan menggunakan pengambilan sampel kepadatan kernel, untuk mendukung parameter mutasi. Perhatikan bahwa meskipun konsep angka inovasi sama dengan NEAT, pengambilan sampel kepadatan kernel merupakan peningkatan tambahan di CATNeuro.

Sistem menggunakan pengambilan sampel acak hingga jumlah nilai ambang batas terkumpul. Saat membuat peta, fitness terbalik dari individu yaitu kerugian pelatihan digunakan sebagai bobot untuk setiap sampel. Normative KS memanfaatkan mekanisme ini untuk memperbarui parameter node grafik dan parameter meta. Perhatikan bahwa Normative KS Topografi menerapkan parameter mutasi secara deterministik, tetapi parameter ini juga dapat dipilih oleh KS lain, sehingga mekanisme pengambilan sampel ini dibagi di antara KS.

KS Topografi mempertahankan gambaran umum yang luas tentang lanskap pencarian dengan mengelompokkan individu-individu terbaik ke dalam K klaster berbeda dengan algoritma pengelompokan K-means. Ide ini terinspirasi oleh algoritma *Brainstorm Optimization* (BSO). K-means membutuhkan ukuran jarak. Metode crossover untuk menyelaraskan dua grafik adalah dasar dari ukuran jarak grafik-ke-grafik yang digunakan untuk K-means. KS Topografi memberi peringkat kepada pemain terbaik dari generasi saat ini dan sebelumnya dan mempertahankan G pemain terbaik. Kemudian mengelompokkannya ke dalam K klaster. Pada saat pengaruh, Topografi secara acak memilih sebuah klaster dan menggunakan grafik centroidnya sebagai dasar untuk mutasi seperti yang dijelaskan sebelumnya. Topografi cenderung bias terhadap crossover seperti yang diilustrasikan pada Gambar 9.6.

9.4 MEKANISME KD

Mekanisme KD mengaitkan KS dengan setiap orang dalam jaringan sosial. Topologi jaringan populasi merupakan masukan ke dalam mekanisme KD. Keadaan tetangga jaringan berpengaruh pada penugasan KS individu. Mekanisme KD merupakan bidang pembelajaran yang aktif. Yang paling banyak digunakan adalah Mayoritas Tertimbang atau WTD. Baru-baru ini, lelang telah diteliti. Permainan Stag-Hunt dari teori permainan evolusioner adalah dasar dari mekanisme KD yang digunakan di sini. Stag-Hunt memungkinkan kerja sama antar KS sedangkan mekanisme lain mendukung persaingan antar KS. Untuk menetapkan apakah kerja sama mendukung tugas NAS yang kompleks lebih baik daripada yang secara inheren kompetitif, kinerja WTD dibandingkan dengan Stag-Hunt. Kedua mekanisme yang digunakan dalam artikel ini dijelaskan selanjutnya.

Mayoritas Tertimbang KD

Mekanisme Distribusi Mayoritas Tertimbang (WTD) menggunakan informasi tentang KS tetangganya untuk menentukan KS bagi individu dalam satu generasi. Ini mirip dengan pemungutan suara alias "kebijaksanaan orang banyak". Optimasi CATNeuro diajukan sebagai masalah minimisasi untuk menemukan jaringan dengan kerugian pelatihan terendah. Misalkan $fit = 1/$(kerugian pelatihan) dan $K = H, S, N, T, D$ adalah himpunan KS. Misalkan i mengindeks individu populasi. $W_{k \in K} = \frac{1}{\sum_i fit_i} \sum_i fit_i \mid k_i = k$ adalah kepentingan relatif (bobot) setiap KS dalam populasi. Misalkan j mengindeks tetangga jaringan i, maka $k_{i1}, k_{i2}, \ldots, k_{ij}$, adalah KS mereka masing-masing. Juga, misalkan $k_{ig} \in K$ adalah KS yang dipilih secara acak dengan probabilitas pemilihan yang proporsional dengan W_k yang disebut pengaruh langsung. KS yang ditugaskan ke i sebelum pengaruh adalah

$$k_{inew} = \left(\sum_j w_{k_{ij}} \mid k_{ij} = k \right) + \begin{cases} w_{k_{ig}} & k_{ig} = k \\ 0 & otherwise \end{cases}$$

dalam WTD, ada kemungkinan KS dihilangkan dari populasi. Jika demikian, KS tersebut dimasukkan kembali dengan menugaskannya kepada sebagian kecil individu yang dipilih secara acak dalam populasi.

Stag-Hunt KD

Dalam "Perburuan Rusa", para pemburu memulai dengan bekerja sama untuk berburu rusa — tugas yang menantang yang menghasilkan hadiah besar. Jika rusa (yang lebih langka) tidak terlihat untuk sementara waktu, para pemburu tergoda untuk "membelot" dan berburu kelinci sebagai gantinya (yang berlimpah). Para pemburu dapat berburu kelinci secara individual tanpa memerlukan kerja sama dari pemburu lain, tetapi hadiahnya jauh lebih rendah.

Perburuan Rusa memodelkan situasi di mana kerja sama adalah tindakan awal yang dipilih oleh para pemain, tetapi masing-masing dapat memilih untuk membelot (bersaing) ketika hadiah yang diharapkan cukup tertunda. Di antara skenario lain, Perburuan Rusa telah digunakan untuk mempelajari pembentukan aliansi dan evolusi struktur sosial. Dalam adaptasi CATNeuro dari Perburuan Rusa, terdapat $P\ (>1)$ generasi kerja sama diikuti oleh satu generasi evaluatif. Siklus ini diulang sampai berakhir. Semua individu bekerja sama dalam generasi kooperatif dan bertindak secara "individualistis" dalam generasi evaluatif.

Dalam Stag-Hunt KD, parameter K, i, dan j didefinisikan seperti sebelumnya di bagian WTD. $F = \{r \mid r \in R\}$ adalah himpunan bilangan riil dan $F_i \subseteq F = \{fit_{i1}, fit_{i2}, \dots, fit_{ij}\}$ adalah kebugaran tetangga i dalam generasi saat ini. SocialRank: $R \times F \rightarrow Z$ adalah fungsi yang mengembalikan peringkat "sosial" i berdasarkan kebugarannya dan nilai kebugaran tetangganya. SocialRank individu i adalah peringkatnya di antara tetangga lokalnya dalam hal kebugaran. Individu i diberi peringkat tinggi jika fiti tinggi dibandingkan dengan tetangganya (dan sebaliknya). Misalkan sr_i SocialRank (fit_i, F_i) adalah peringkat sosial i. Juga, RankKS: $Z \rightarrow K$ mengembalikan KS berdasarkan SocialRank. Kemudian, $k_{inew} = \mathrm{RankKS}(sr_i)$. Fungsi RankKS mengembalikan KS yang relatif eksploratif ketika peringkatnya rendah dan yang eksploitatif sebaliknya. Mekanisme ini dengan demikian mengaitkan strategi eksploratif dengan individu yang berkinerja relatif rendah dan sebaliknya. Keseimbangan eksplorasi-eksploitasi dipertahankan menggunakan sinyal lokal dari peringkat "sosial".

Dalam generasi yang evaluatif, misalkan $O =< k_0, k_1, \dots, k_{n-1} >$ adalah urutan KS menurut potensi eksploratif masing-masing; o_k adalah offset k dalam urutan ini dan $O[q]$ mengembalikan k pada offset q. Misalkan $e_i, e_2, \dots, e_g, \dots$ mengindeks generasi evaluatif dan fit_{ie_g} adalah kebugaran i dalam generasi eg. Maka,

$$k_{inew} = \begin{cases} O\left[O_{k_i} - 1 \; mod \; n\right] & fit_{ie_g} \geq fit_{ie_{g-1}} \\ O\left[O_{k_i} + 1 \; mod \; n\right] & \text{otherwise.} \end{cases}$$

Singkatnya, jika kebugaran i telah meningkat sejak generasi evaluasi sebelumnya maka ia mengadopsi KS yang paling eksploitatif berikutnya dalam tangga (dan berputar kembali jika berada di bawah) dan sebaliknya. Oleh karena itu, i bertindak secara individual tanpa mempertimbangkan tetangganya.

Spesiasi

Seperti halnya Miikkulainen, CATNeuro juga mengimplementasikan spesiasi, yaitu beberapa populasi dalam satu sistem. Terdapat satu blueprint dan satu atau lebih spesies modul. Setiap spesies digerakkan oleh proses CA independen (misalnya memiliki Ruang Kepercayaan sendiri, dll.). Semuanya dikonfigurasi dengan mekanisme KD yang sama. Node terminal dari grafik blueprint adalah jangkar untuk data input dan output jaringan saraf. Setiap node internal dari grafik blueprint menunjuk ke spesies modul. Grafik spesies modul terstruktur sebagai subgraf yang dapat disematkan. Node grafik internal dari individu spesies modul mewakili lapisan jaringan saraf, misalnya. CNN, normalisasi (normalisasi batch atau layer), lapisan dense, dll. Pada saat evaluasi, model jaringan saraf abstrak dirakit dari grafik cetak biru dengan mengganti node internal dengan anggota yang dipilih secara acak dari spesies modul yang dirujuk. Jika beberapa node dari cetak biru menunjuk ke spesies yang sama, semuanya diganti dengan anggota yang sama dari spesies tersebut. Hal ini mendorong pengembangan komponen modular dan dapat digunakan kembali.

Banyak arsitektur jaringan baru menggunakan "sel" sebagai blok bangunan (misalnya ResNet atau BERT). Oleh karena itu, pemilihan anggota modul dilakukan untuk membantu penemuan komponen modular. Selanjutnya, model yang dirakit (tetapi masih abstrak) diterjemahkan ke dalam model konkret yang menargetkan kerangka kerja DL tertentu, misalnya CNTK atau PyTorch. Model konkret kemudian dievaluasi oleh antarmuka eksternal yang mengembalikan kerugian dan ukuran model (setelah melatih dan mengevaluasi model pada set pengujian). Individu cetak biru diberi nilai kerugian yang dikembalikan sebagai nilai kebugaran. Individu modul diberi nilai kebugaran rata-rata dari individu cetak biru di mana anggota tersebut digunakan. Ini melindungi anggota spesies modul agar tidak dieliminasi terlalu dini sementara nilainya masih ditetapkan.

Pemangkasan Ruang Pencarian

CATNeuro memungkinkan pembatasan diterapkan pada tingkat spesies. Pertama, jumlah node dapat dibatasi hingga maksimum. Kedua, operasi tertentu dapat dikecualikan. Sistem akan mengalihkan sumber daya ke operasi grafik lain sebagai gantinya di bawah pembatasan tersebut. Ketiga, jumlah parameter untuk berbagai operasi jaringan saraf (misalnya lapisan padat) atau hiperparameter dapat dibatasi pada rentang yang diinginkan. Dalam eksperimen yang dilakukan di sini, satu spesies modul didedikasikan untuk menemukan representasi yang dipelajari sebagai embedding. Pada spesies ini, jumlah maksimum node dibatasi menjadi satu node internal (tiga total termasuk input dan output) dan operasi grafik dibatasi hanya pada parameter yang dimutasi.

9.5 MASALAH BENCHMARK GAME 2D

Masalah benchmark yang digunakan untuk membandingkan WTD dan Stag-Hunt adalah game pertarungan 2D bernama FightingICE (Gambar 9.8). FightingICE adalah platform uji coba pembelajaran untuk AI yang dikelola oleh *Intelligent Computer Entertainment* (ICE) Lab, Universitas Ritsumeikan, Jepang. ICE juga mengadakan kompetisi tahunan untuk game tersebut.

Kerangka kerja ICE dirancang untuk penyematan pengontrol AI kompetitif. Status game disiarkan ke setiap pemain di setiap frame. Pengontrol AI menggunakannya untuk pengambilan keputusan. Namun, data frame tertunda selama 15 frame, sehingga AI harus prediktif. Setiap pemain memiliki 56 kemungkinan aksi yang tersedia dapat dibagi menjadi serangan, pertahanan, dan penempatan posisi. AI setiap pemain diharapkan memberikan aksi untuk dieksekusi oleh mesin game, ketika diminta.

Simulator game juga disediakan untuk FightingICE. Simulator menerima frame game yang valid dan dua urutan aksi, satu untuk setiap pemain. Simulator ini dapat digunakan untuk mengevaluasi serangkaian gerakan yang dimulai dari status game tertentu. Simulator ini dimanfaatkan secara intensif di sini untuk mempelajari kebijakan tindakan dengan pembelajaran penguatan (*reinforcement learning*), yang akan dijelaskan selanjutnya.



Gambar 9.8 Tangkapan layar FightingICE.

Permainan 2D: Pembelajaran Penguatan untuk Membuat Data Pelatihan

Data yang dibutuhkan untuk melatih dan mengembangkan model DL dibuat dengan bantuan pembelajaran penguatan. Tugas penting bagi pengontrol adalah memilih salah satu dari 56 tindakan yang tersedia di setiap frame berdasarkan informasi status permainan yang tersedia ditambah beberapa riwayat. Tujuannya adalah untuk memberikan pukulan maksimal pada lawan sambil melindungi diri sendiri, di setiap ronde. Pengaturan seperti itu disebut Proses Keputusan Markov (juga Proses Hadiah Markov) dan umumnya digunakan di bidang robotika, sistem multi-agen, permainan, dan aplikasi kontrol.

RL adalah bidang studi yang luas yang mencakup berbagai metode. Di sini, pendekatan RL berbasis kebijakan digunakan. Tabel kebijakan dipelajari yang menjadi data pelatihan untuk jaringan saraf. Tabel tersebut memetakan vektor 74 × 1 yang mewakili status permainan ke vektor 56 × 1 yang mewakili distribusi probabilitas atas kumpulan tindakan yang tersedia. Vektor status permainan berisi data relevan dari frame permainan saat ini dan sebelumnya,

seperti posisi dan kecepatan pemain, skor dan tingkat energi mereka, riwayat tindakan terbaru mereka, dll.

Pertama, banyak permainan dimainkan menggunakan AI sampel dan semua frame permainan disimpan. Kedua, setiap frame yang disimpan kemudian digunakan dengan simulator untuk memainkan cuplikan permainan pendek dengan urutan tindakan yang berbeda. Proses ini digunakan untuk membangun tabel kebijakan sesuai dengan metodologi RL. Tabel kebijakan memiliki 1.3 juta entri dan menghabiskan sekitar 1 GB ruang disk. Menarik untuk dicatat bahwa model yang dihasilkan dari data pelatihan yang pada dasarnya merupakan representasi terkompresi dari tabel berukuran kurang dari 100 KB. Lebih lanjut, tabel tersebut merupakan pemetaan diskrit dari status model ke distribusi tindakan. Dengan demikian, tabel tersebut tidak cocok untuk permainan karena semua kemungkinan status tidak terwakili (dan juga tidak mungkin untuk melakukannya). Model DL, sebaliknya, menyediakan pemetaan kontinu (melalui interpolasi), dan karenanya cocok untuk permainan.

Game 2D: Pengaturan Eksperimental

Untuk mengurangi waktu pelatihan jaringan saraf selama menjalankan CATNeuro, sampel sebanyak 130 ribu baris dipilih secara acak dari tabel lengkap berisi 1.3 juta baris. Ukuran populasi yang digunakan adalah 36 (yaitu, 36 pelatihan dan evaluasi jaringan saraf per generasi). Konfigurasi perangkat keras saat ini terbatas pada satu GPU. Dibutuhkan 20–30 menit untuk melatih dan mengevaluasi model untuk satu generasi. Namun, pelatihan dapat diparalelkan, sehingga sumber daya tambahan (misalnya, lingkungan komputasi awan) akan mengurangi waktu ini secara signifikan.

Tabel 9.1 Pengaturan eksperimental permainan 2D CATNeuro.

Parameter	Nilai	Komentar
Perangkat Keras	Laptop gaming Acer Predator GX-792 dengan kartu grafis Nvidia GTX 1080	RAM 32 GB / memori GPU 8 GB, OS Windows 10
Lawan AI	Jerry Mizuno (tingkat menengah) dan Thunder (juara 2018)	
Mekanisme KD	Stag-Hunt, Weighted majority (WTD)	
Ukuran populasi	36	Untuk semua populasi
Topologi jaringan	Heksagonal	
Jumlah spesies modul	3	Catatan: Ukuran populasi dan jumlah spesies sebanding dengan yang digunakan dalam
Batas cetak biru	Maksimal 20 node	
Batas Modul 1	3 node	Hanya mengubah parameter untuk mendukung *embedding*
Batas Modul 2 dan 3	4 node	

Kondisi terminasi	Berhenti jika tidak ada peningkatan yang terlihat dalam 10 generasi	
Dataset (tabel kebijakan)	130 K sampel untuk NAS, 1,3 M untuk pelatihan akhir	*Split* latih-uji 70/30
Ukuran sampel	6 per mekanisme KD	Dapat memakan waktu 24–36 jam untuk selesai
Model dievaluasi dengan pengontrol game	12 per mekanisme KD	Ambil 2 teratas (10%) dari setiap putaran, 6 putaran per mekanisme KD
Pustaka DL	Microsoft CNTK	Dukungan untuk pustaka lain direncanakan
Jumlah game yang dimainkan per model	10	Setiap game memiliki 3 ronde

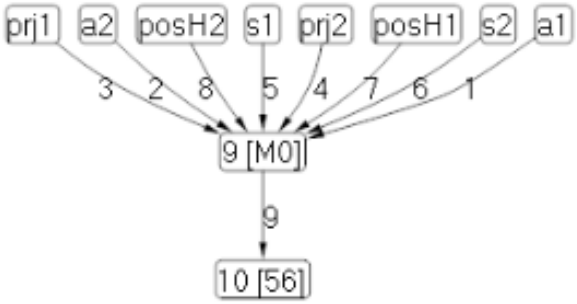
Dataset 130 ribu dibagi 70/30 untuk pelatihan/evaluasi. Model yang dirakit dilatih pada set pelatihan dan diuji pada set evaluasi hingga kerugian evaluasi mulai meningkat (setelah periode pemanasan). Hal ini dilakukan untuk mencegah overfitting. Pelatihan dihentikan jika tidak ada peningkatan yang terlihat selama 10 generasi. CATNeuro berjalan selama 20–60 generasi sebelum dihentikan untuk sebagian besar percobaan.

10% model terbaik diambil dari setiap percobaan dan dilatih pada dataset lengkap 1,3 juta baris, kemudian dijalankan di pengontrol untuk bermain melawan pengontrol AI yang tersedia: (a) ``Jerry Mizuno" — AI yang digerakkan secara algoritmik yang tersedia dengan paket FightingICE yang digunakan sebagai tolok ukur dan (b) ``Thunder" sang juara 2018. Data relevan dari permainan yang dimainkan disimpan ke file log untuk pemeriksaan selanjutnya. Kinerja model yang dihasilkan WTD dan Stag-Hunt dibandingkan untuk kedua lawan. Sebagian dari hasilnya diperiksa di bagian selanjutnya. Pengaturan eksperimental terperinci ada di Tabel 9.1.

Sistem CATNeuro dikonfigurasi untuk menghasilkan 20 model terbaik yang ditemukan selama percobaan, tetapi hanya 2 model terbaik (10% teratas) yang digunakan di pengontrol untuk permainan. Namun, semua 20 model digunakan untuk mempelajari properti model yang dihasilkan oleh kedua mekanisme tersebut. Sebagian dari properti yang dipelajari tercantum dalam Tabel 9.2.

Tabel 9.2 Properti model yang diperiksa.

Metrik Model	Komentar
Loss Pelatihan	*Loss* pelatihan adalah jumlah kuadrat kesalahan antara *output* model dan kebenaran dasar (*ground truth*) dari data pelatihan. Ukuran sampel = 6 * 20 = 120
Jumlah bobot parameter	Jumlah total bobot dalam model DL yang dioptimalkan dalam pelatihan. Ini adalah ukuran dari besarnya model. Ukuran sampel: 120
Panjang jalur maksimum	Panjang jalur terpanjang di setiap model, dari *node* input ke *node* output.



Gambar 9.9 Grafik cetak biru "pemula" tipikal ketika populasi diinisialisasi.

Gambar 9.9 menunjukkan contoh grafik tipikal ketika populasi CA diinisialisasi. Grafik awal adalah struktur sederhana. Gambar 9.9 adalah grafik individu cetak biru tipikal pada inisialisasi populasi. Input berada di atas dan output di bawah. Vektor input status permainan 74 × 1 dibagi menjadi delapan unit "semantik" terpisah yang mengelompokkan nilai-nilai serupa untuk memfasilitasi pembuatan representasi tertanam. Ini untuk posisi 2D (x dan y), kecepatan 2D (x dan y), tingkat energi, dan riwayat aksi untuk masing-masing dari dua lawan. Node internal mengacu pada spesies Modul 0 ("M0"). Setelah perakitan, node ini akan berisi subgraf dari individu M0 yang dipilih secara acak. Angka-angka yang terkait dengan setiap tautan adalah angka inovasi. Populasi modul dan cetak biru diinisialisasi dengan struktur grafik sederhana.

Permainan 2D: Hasil Eksperimental

Model yang dihasilkan oleh WTD dan Stag-Hunt dimainkan melawan Jerry Mizuno dan "Thunder". Statistik ringkasan untuk Jerry Mizuno ada di Tabel 9.3. Dalam hal "pukulan ke lawan" per ronde, kedua mekanisme KD menghasilkan model yang dengan mudah menang melawan Jerry Mizuno, AI benchmark yang disertakan dalam paket. Berbeda dengan Thunder, Jerry Mizuno sebagian besar digerakkan secara algoritmik. Ia menggunakan K-nearest neighbors dan metode logika fuzzy dan karenanya lebih sebanding dengan pendekatan berbasis RL yang digunakan di sini. Namun, tidak ada perbedaan statistik antara kinerja versi WTD dan Stag-Hunt karena keduanya hanya membutuhkan perilaku dasar untuk mengatasi lawan AI tersebut.

Tabel 9.4 menunjukkan hasil yang setara untuk lawan yang jauh lebih kuat, Thunder — juara tahun 2018. Thunder dengan mudah mengalahkan kedua model yang dihasilkan oleh CATNeuro. Namun, di sinilah model yang dihasilkan oleh Stag-Hunt secara statistik lebih baik daripada model untuk WTD. Perlu dicatat bahwa tidak seperti Jerry Mizuno, Thunder dikodekan secara manual oleh seorang ahli manusia. AI ini memiliki pengetahuan tentang aturan permainan, eksploitasi, dan kombinasi spesifik yang sulit dipelajari oleh pendekatan berbasis RL murni karena keterbatasan sumber daya komputasi. Dalam satu eksploitasi spesifik, Thunder mampu memojokkan lawan dengan gerakan yang tepat waktu sehingga lawan tidak dapat pulih dengan cepat.

Tabel 9.3 Ringkasan kinerja — CATNeuro versus Jerry Mizuno Opp.

	Rata-rata serangan ke lawan per ronde	
	Stag-Hunt	WTD
Rata-rata serangan/ronde	25.63	25.64
Standar deviasi	4.88	4.46
Nilai p uji-t dua sampel	0.987 (sampel = 360)	

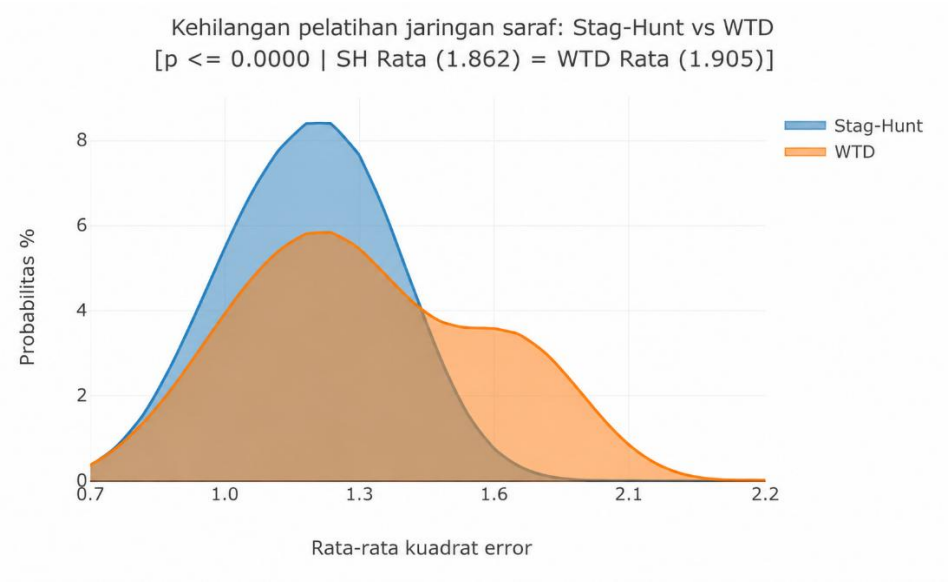
Tabel 9.4 Ringkasan kinerja — CATNeuro versus Thunder Opp.

	Rata-rata serangan ke lawan per ronde	
	Stag-Hunt	WTD
Rata-rata serangan/ronde	10.48	9.38
Standar deviasi	4.25	3.91
Nilai p uji-t dua sampel	0.0003 (sampel = 360)	

Selain itu, Thunder menggunakan gerakan "kombo" di mana poin maksimum dapat diekstraksi dari lawan, mengingat aturan permainan tertentu. Untuk pendekatan RL gradien kebijakan untuk "mengungkap" aturan kompleks seperti itu membutuhkan sumber daya komputasi yang jauh lebih besar daripada yang tersedia. Sebagai referensi, pertimbangkan bahwa AlphaGo Zero awalnya membutuhkan 40 hari pelatihan dengan 64 GPU (1920 GPU awalnya) untuk mencapai kinerja superhuman. Kuncinya adalah AlphaGo tidak perlu mempelajari aturan dasar Go. Aturan-aturan tersebut diprogram oleh manusia.

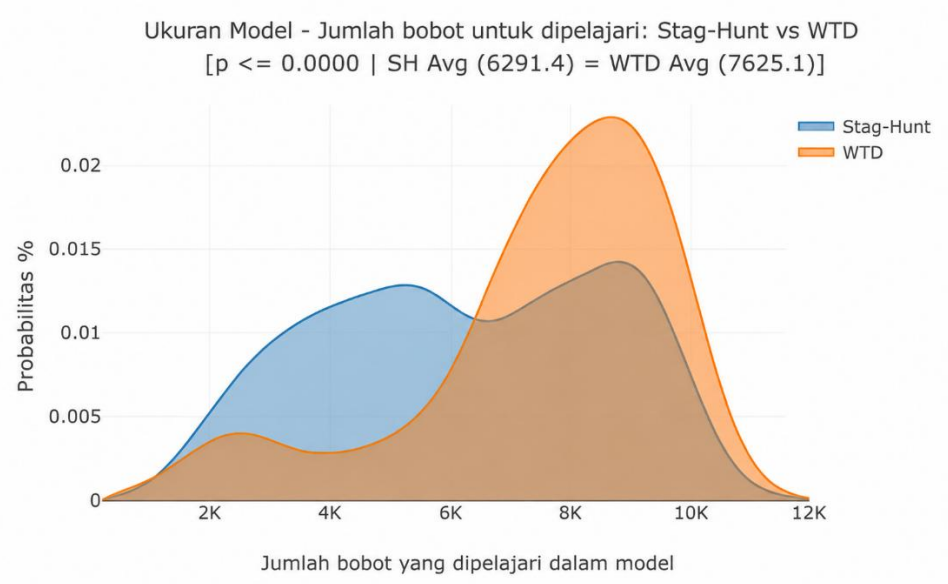
Meskipun mekanisme Stag-Hunt mengungguli pendekatan Weighted Majority dalam hal kinerja, area menarik lainnya adalah kualitas model yang dihasilkan antara kedua KD tersebut. Setiap eksekusi CATNeuro mengembalikan 20 model teratas yang ditemukan dalam eksekusi tersebut. Karena ada enam eksekusi per mekanisme, ada 6 × 20 = 120 model untuk masing-masing mekanisme. Kedua set ini digunakan untuk membandingkan kualitas struktural model yang dihasilkan oleh kedua pendekatan tersebut. Meskipun sebagian besar model ini tidak digunakan untuk bermain game, model-model ini masih dapat berfungsi sebagai dasar perbandingan antara mekanisme Stag-Hunt dan WTD. Semua model yang dihasilkan merupakan kandidat untuk bermain melawan kedua lawan, meskipun hanya model teratas dari masing-masing mekanisme yang digunakan dalam pengontrol.

Mempertimbangkan sifat-sifat model yang dihasilkan, Gambar 9.10 berisi plot kepadatan yang ditumpangkan untuk metrik kerugian pelatihan untuk semua 120 model yang dihasilkan oleh setiap mekanisme. Sumbu X mewakili kerugian pelatihan dalam hal kesalahan kuadrat rata-rata. Sumbu Y adalah probabilitas kerugian tersebut. Luas setiap kurva berjumlah 1. Kurva menunjukkan distribusi massa (probabilitas) terhadap kehilangan pelatihan untuk mekanisme KD terkait. Nilai P (≤0.00001) untuk perbedaan rata-rata diberikan di area judul. Grafik menunjukkan bahwa mekanisme KD Stag-Hunt menghasilkan model yang memiliki kehilangan pelatihan yang jauh lebih rendah. Meskipun kedua mekanisme memiliki puncak pusat yang serupa, pendekatan WDT memiliki puncak sekunder dengan kehilangan pelatihan yang lebih tinggi.



Gambar 9.10 Distribusi kerugian pelatihan model berdasarkan KD.

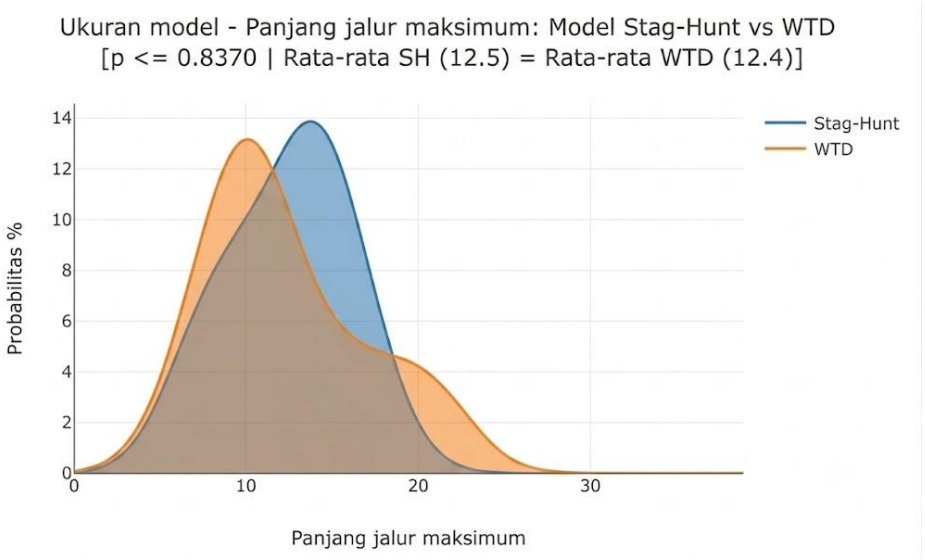
Gambar 9.11 membandingkan kepadatan ukuran model antara model yang dihasilkan oleh kedua mekanisme. Stag-Hunt KD lebih mungkin menemukan model dengan parameter yang lebih sedikit untuk dipelajari daripada mekanisme Weighted Majority KD. Disarankan bahwa mekanisme kooperatif lebih mampu mengkonsolidasikan dan menggabungkan bagian-bagian model yang tidak dapat dilakukan oleh pendekatan yang sepenuhnya kompetitif.



Gambar 9.11 Distribusi jumlah parameter yang dapat dipelajari (yaitu ukuran model) berdasarkan KD.

Gambar 9.12 membandingkan distribusi panjang jalur maksimum pada model yang dihasilkan oleh kedua mekanisme KD. Di sini, perbedaan rata-rata tidak signifikan secara statistik antara kedua distribusi. Namun, bentuk kurva distribusi menunjukkan bahwa Stag-

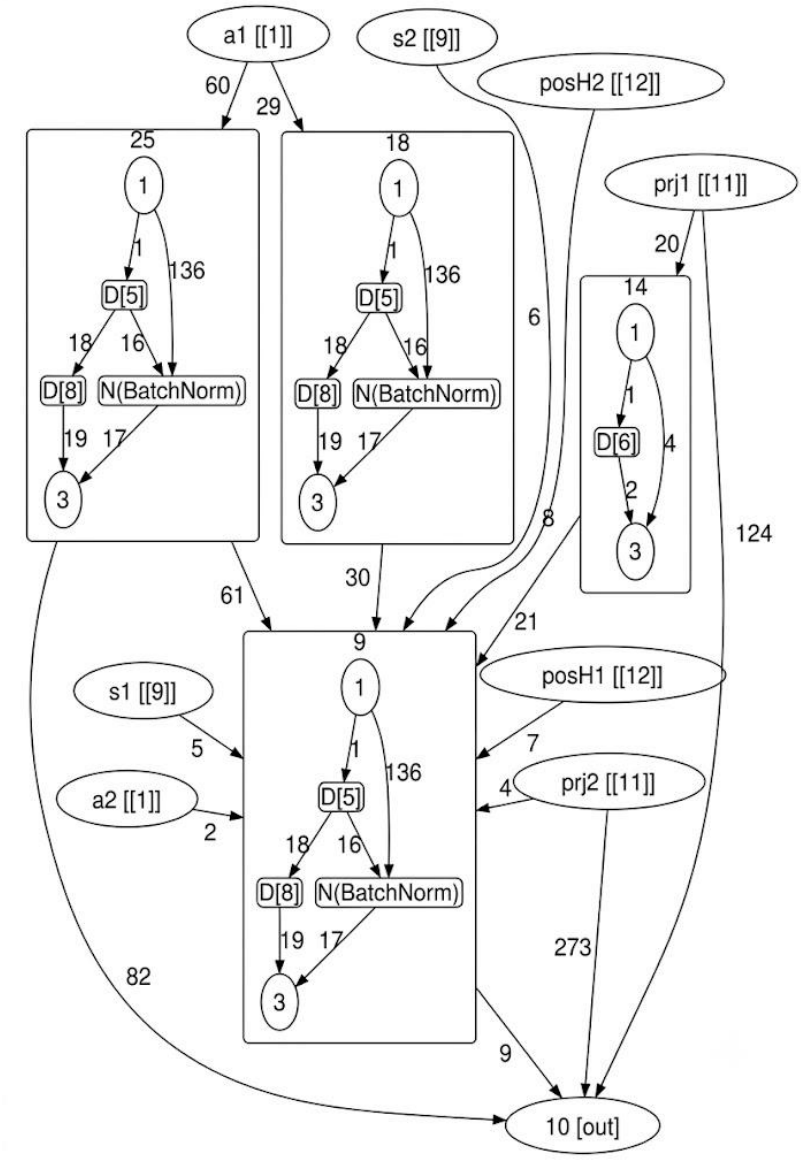
Hunt lebih sering menghasilkan model yang lebih dalam (modus untuk Stag-Hunt lebih tinggi). Selain itu, model berkinerja terbaik yang dihasilkan oleh masing-masing mekanisme KD menyoroti faktor ini.



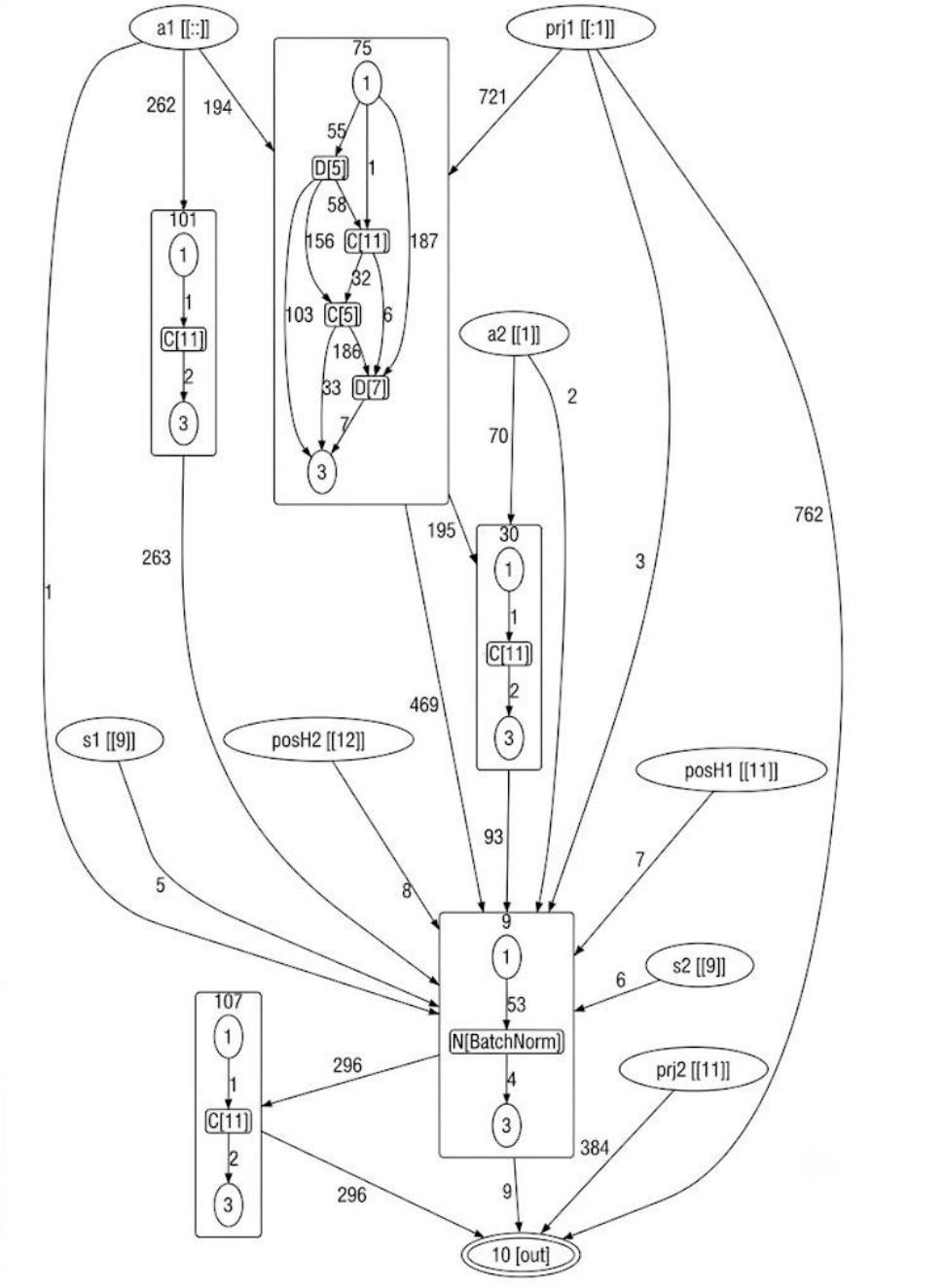
Gambar 9.12 Distribusi panjang lintasan maksimum pada model yang dihasilkan oleh setiap KD.

Gambar 9.13 dan 9.14 adalah struktur model terbaik yang dihasilkan oleh WTD dan Stag-Hunt, masing-masing. Oval garis tunggal mewakili node input dan oval garis ganda mewakili output. Persegi panjang besar mewakili node modul, dan subgraf yang terdapat di dalamnya adalah individu dari spesies modul. Node di dalam modul diberi kode "D" untuk dense (padat) dan "N" untuk normalisasi.

Perhatikan bahwa vektor input 74 × 1 dipecah menjadi delapan unit "semantik" (oval garis tunggal). Unit semantik ini (x dan y), kecepatan (x dan y), tingkat energi, riwayat tindakan, dll. dikelompokkan ke dalam node input terpisah untuk memfasilitasi pembelajaran representasi tersemat. Sudah diketahui bahwa jaringan saraf, terutama dalam bentuk model DL kontemporer, mempelajari representasi perantara dari data input sebagai embedding. Seringkali, representasi tersebut secara signifikan meningkatkan kinerja dan merupakan fitur standar dari model kontemporer. Alih-alih memasukkan vektor 74 × 1 sebagai input tunggal, input dikelompokkan ke dalam vektor terpisah dari nilai-nilai terkait sehingga akan lebih mudah bagi model untuk mempelajari representasi tersemat. Ini karena dalam neuroevolusi, model sederhana dievaluasi terlebih dahulu.



Gambar 9.13 Struktur model terbaik WTD berdasarkan "pukulan ke lawan" [Training Loss = 1,91, Tunable Parms = 9619, Nodes = 27, Conns = 37, Depth = 9].



Gambar 9.14 Struktur model terbaik Stag-Hunt berdasarkan "pukulan ke lawan" [Training Loss = 1,85, Tunable Parms: 9556, Nodes = 27, Conns = 39, Depth = 16].

Model akhir dikembangkan melalui kompleksifikasi progresif dari model yang lebih sederhana ini. Struktur model "terbaik" yang ditunjukkan di sini mungkin tampak tidak konvensional dari sudut pandang manusia tetapi valid dalam hal kinerja di luar sampel. Model Stag-Hunt jauh lebih dalam (kedalaman = 16) daripada WTD (kedalaman = 9), namun sedikit lebih kecil dengan 9556 parameter dibandingkan dengan 9619 parameter WTD. Selain itu, model Stag-Hunt memiliki kerugian yang lebih rendah yaitu 1.85 MSE dibandingkan dengan WTD (1.91). Secara umum, model yang lebih dalam memiliki kapasitas pembelajaran yang lebih besar.

Pernyataan ini didukung dengan baik di bagian 6.4.1 dari buku Goodfellow dkk. "DL". Karena Stag-Hunt cenderung menemukan model yang lebih dalam lebih sering (Gambar 9.12), hal ini menunjukkan bahwa alokasi atau campuran strategi pencarian yang lebih optimal diberikan daripada mekanisme pemungutan suara yang sesuai seperti WDT untuk tugas-tugas kompleks seperti NAS.

9.6 MASALAH BENCHMARK PELACAKAN GERAK TRAILER

Contoh sebelumnya mengilustrasikan dampak yang dapat ditimbulkan oleh mekanisme KD pada pemecahan masalah dalam mesin evolusi berbasis jaringan seperti CA. Perbandingan kedua mekanisme distribusi menunjukkan kinerja dan keunggulan struktural dari mekanisme kooperatif-kompetitif dibandingkan dengan mekanisme yang sepenuhnya kompetitif untuk masalah desain kompleks yang disajikan. Pada bagian ini, konfigurasi Stag-Hunt CATNeuro yang paling sukses dari keduanya akan digunakan untuk menunjukkan sinergi yang muncul dengan menggunakan algoritma evolusi berbasis pengetahuan untuk memandu proses neuroevolusi NEAT.

Sebuah traktor-trailer adalah sistem dinamis yang kompleks seperti halnya pengontrol game. Ada bahaya sistem menjadi tidak stabil dalam kondisi kecepatan, distribusi berat, dan percepatan longitudinal/lateral/angular tertentu. Misalnya, trailer dapat bergoyang yang dapat membahayakan stabilitas seluruh sistem. Traktor dapat mengkompensasi (misalnya dengan menerapkan pengereman diferensial) untuk mengontrol atau meredam goyangan trailer. Namun, semua mekanisme kontrol tersebut memerlukan perkiraan massa trailer. Sebagian besar trailer tidak dilengkapi instrumen untuk menyampaikan informasi massa ke sistem kontrol. Alternatifnya adalah memperkirakan massa dari kendaraan traktor (yang jauh lebih baik dalam hal instrumentasi) menggunakan metrik seperti berat traktor, torsi mesin, laju akselerasi, dan posisi throttle.

Kemampuan untuk melacak pergerakan trailer dari waktu ke waktu dapat lebih meningkatkan akurasi estimasi massa dengan asumsi faktor lain tetap sama, trailer yang lebih berat akan menunjukkan osilasi frekuensi yang lebih rendah daripada yang lebih ringan. Selain itu, deteksi peningkatan amplitudo goyangan trailer dapat berfungsi sebagai peringatan dini terhadap ketidakstabilan sistem. Pembahasan model dinamis untuk estimasi massa berada di luar cakupan artikel ini. Tugas di sini adalah membangun jaringan saraf untuk mengukur pergerakan trailer secara akurat dalam arah lateral dan vertikal menggunakan gambar dari kamera cadangan murah yang dipasang pada kendaraan traktor.

Pergerakan Trailer: Data Pelatihan

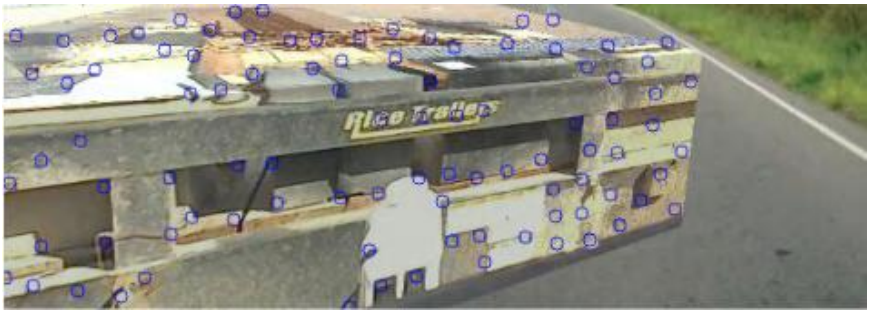
Masalah utama dalam melatih model seperti itu adalah ketersediaan data pelatihan berlabel. Akan sangat sulit bagi manusia untuk memberi label yang benar pada rekaman aktual trailer yang ditarik. Untuk mengatasi masalah ini, dunia virtual "trailer" tiga dimensi (3D) dibuat seperti yang diilustrasikan pada Gambar 9.15. Adegan tersebut menunjukkan model trailer kotak yang dilapisi dengan gambar trailer sebenarnya.

Kamera adegan 3D diposisikan dan diorientasikan seolah-olah ditempatkan di bagian belakang kendaraan traktor. Sistem secara terprogram "menggerakkan" trailer ke kiri/kanan,

atas/bawah, dan secara berkala menyimpan cuplikan tampilan kamera bersama dengan posisi trailer sebenarnya (yang diwakili oleh titik tengah di bagian depan trailer). Sistem menggerakkan trailer seolah-olah terpasang pada titik pengait di traktor. Di sini, posisi trailer sebenarnya diketahui karena dikendalikan oleh sistem yang menempatkan trailer di adegan 3D. Cuplikan yang disimpan diproses lebih lanjut untuk membuat data pelatihan masukan dalam dua langkah. Pertama, gambar dipotong ke wilayah bawah 800 W × 240 H. Kedua, deteksi sudut Harris digunakan untuk menemukan fitur yang akan dilacak pada gambar yang dipotong (Gambar 9.16).



Gambar 9.15 Adegan virtual 3D dari model trailer "kotak" yang dilapisi dengan gambar trailer sebenarnya.



Gambar 9.16 Cuplikan yang dipotong dengan "sudut" yang ditandai dengan algoritma deteksi sudut Harris.

Bingkai baru berukuran 800 × 240 dibuat di mana piksel yang berisi sudut mendapat nilai +1,0 dan -1,0 untuk piksel lainnya. Dua frame berurutan dipasangkan untuk membuat "volume" berukuran 2 × 800 × 240. Jarak sebenarnya antara dua frame berurutan dalam arah x (lateral) dan y (vertikal) dihitung dari label posisi yang disimpan sebelumnya. Dengan demikian, dataset pelatihan model terdiri dari volume $\{+1, -1\}^{2\times800\times240}$ yang masing-masing dipasangkan dengan vektor 2 × 1 yang sesuai dari jarak (x, y) dalam piksel pecahan. Sekitar 10 ribu gambar disimpan. Setiap gambar terdiri dari dua skin trailer dan tiga latar belakang yang dicampur. Pembagian data latih/uji ditetapkan menjadi 90/10 persen.

Perlu dicatat bahwa metode aliran optik klasik seperti Lucas–Kanade dipertimbangkan tetapi ditemukan tidak cukup presisi untuk masalah ini. Sistem pelacakan aliran optik yang sangat presisi (dan mahal) memang ada. Namun, sistem tersebut terutama untuk penggunaan laboratorium. Tujuan dari sistem CATNeuro di sini adalah untuk mencapai akurasi tinggi dengan kamera cadangan yang murah. "Jarak" yang akan diprediksi adalah dalam piksel, sehingga beberapa kalibrasi tambahan akan diperlukan dalam aplikasi produksi untuk mengonversinya ke meter atau satuan standar lainnya. Selain itu, aplikasi produksi kemungkinan besar harus menangani banyak aspek lain yang tidak termasuk dalam ruang lingkup artikel ini.

Gerakan Trailer: Pengaturan Eksperimental

Tugasnya adalah mengembangkan model yang menerima tensor 2 × 800 × 240 dan menghasilkan vektor 2 × 1. Kesalahan kerugian kuadrat terhadap kebenaran dasar harus diminimalkan. CATNeuro yang dikonfigurasi dengan Stag-Hunt KD dibandingkan dengan pendekatan evolusi elitis NEAT. Detail pengaturan eksperimental diberikan pada Tabel 9.5.

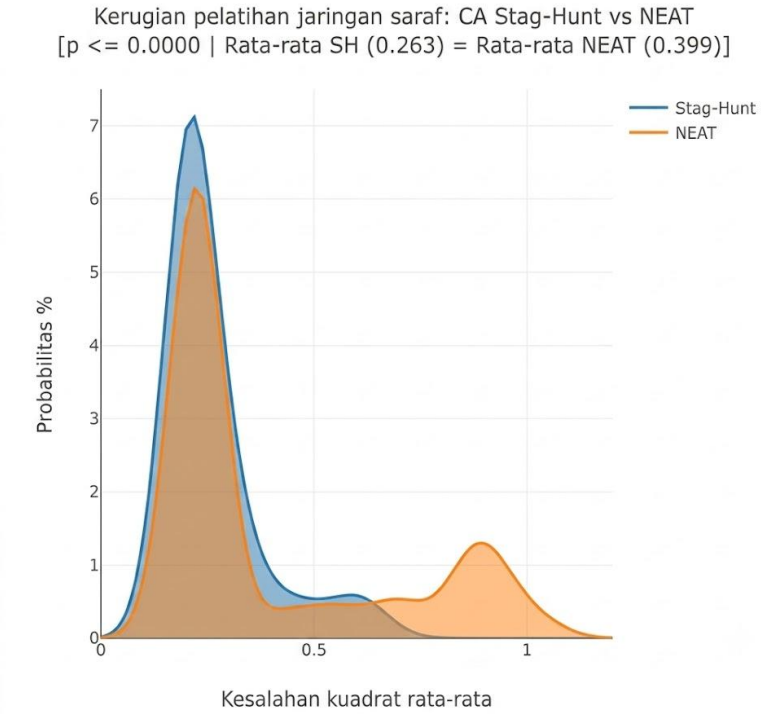
Tabel 9.5 Pengaturan eksperimen gerakan trailer.

Parameter	Nilai	Komentar
Perangkat Keras	Laptop gaming Acer Predator GX-792 dengan kartu grafis Nvidia GTX 1080	RAM 32 GB / memori GPU 8 GB, OS Windows 10
Penggerak Evolusioner	CA dengan Stag-Hunt	Untuk NEAT, 15% individu teratas digunakan untuk mengevolusi populasi berikutnya
Ukuran populasi	36	Untuk semua populasi
Topologi jaringan	Heksagonal	
Jumlah spesies modul	3	Catatan: Ukuran populasi dan jumlah spesies sebanding dengan yang digunakan dalam [10]
Batas cetak biru	Maksimal 12 node	
Batas Modul 1	3 node	Hanya mengubah parameter untuk mendukung *embedding*
Batas Modul 2 dan 3	4 node	
Kondisi terminasi	Berhenti jika tidak ada peningkatan yang terlihat dalam 50 generasi	
Dataset (volume "sudut" berlabel)	10 K untuk NAS dan pelatihan akhir	*Split* latih-uji 90/10
Ukuran sampel	6 per mekanisme	
Pustaka deep learning	Microsoft CNTK	Dukungan untuk pustaka lain direncanakan

Gerakan Trailer: Hasil Eksperimental

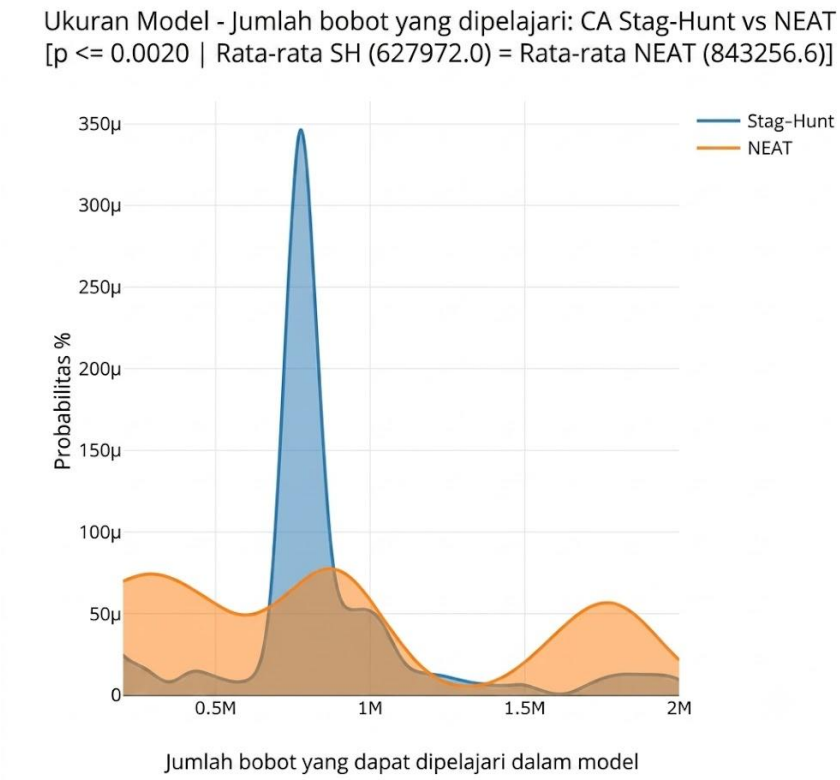
Beberapa metrik dihitung untuk membandingkan kedua mekanisme tersebut. Hanya metrik dengan perbedaan kinerja yang signifikan secara statistik yang dibahas di sini. Gambar 9.17 menunjukkan kerugian pelatihan model untuk model terbaik yang dihasilkan dalam setiap

percobaan. CA dengan Stag-Hunt menghasilkan model dengan kerugian rata-rata yang lebih rendah daripada NEAT (0.263 versus 0.399). Lebih lanjut, distribusi menunjukkan bahwa kerugian pelatihan CATNeuro dengan Stag-Hunt terdistribusi lebih sempit daripada kerugian untuk NEAT.



Gambar 9.17 Distribusi kerugian pelatihan berdasarkan model yang dihasilkan oleh masing-masing mekanisme.

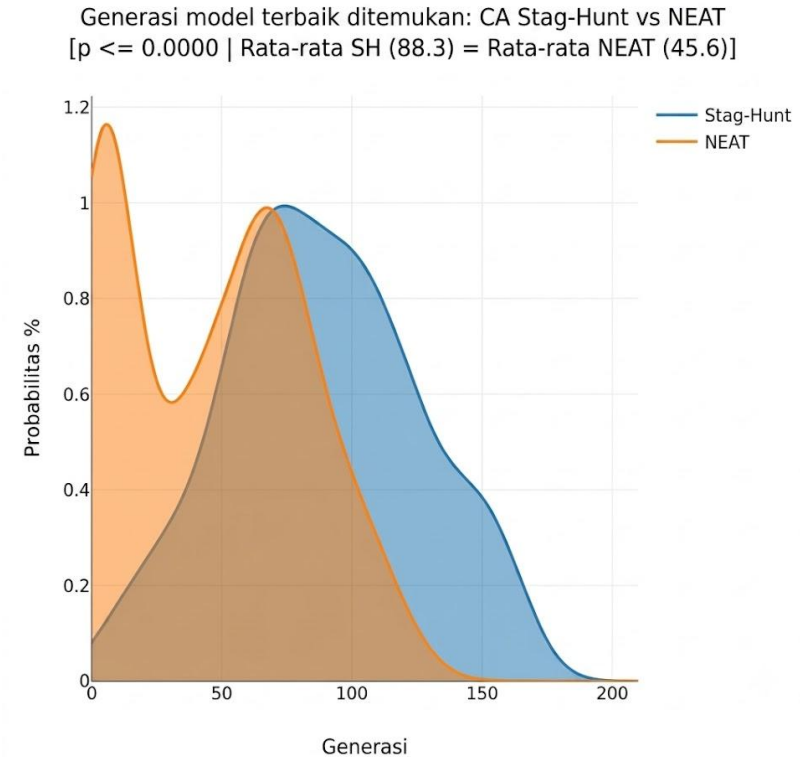
Gambar 9.18 membandingkan ukuran model antara kedua pendekatan tersebut. Terdapat perbedaan yang mencolok antara Stag-Hunt dan NEAT secara statistik. Model yang dihasilkan oleh Stag-Hunt tidak hanya lebih kecil (rata-rata), tetapi distribusinya juga jauh lebih sempit daripada model untuk NEAT. Ini menunjukkan bahwa Stag-Hunt dapat mempertahankan keragaman model pada generasi awal dan kemudian fokus pada struktur yang tepat pada generasi selanjutnya. NEAT "terkunci" di awal dan kehilangan keragaman di kemudian hari.



Gambar 9.18 Distribusi ukuran model — dalam hal bobot yang dapat dipelajari — berdasarkan model yang dihasilkan oleh setiap mekanisme.

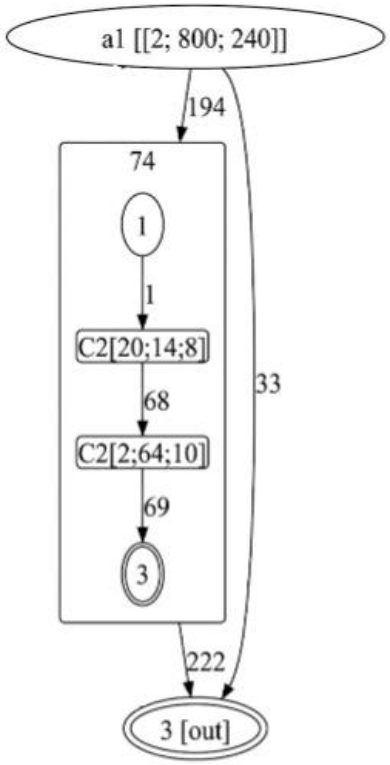
Hal ini juga dibuktikan oleh Gambar 9.19 Gambar tersebut menunjukkan distribusi generasi ketika model terbaik menemukan nilai untuk kedua pendekatan. CATNeuro dengan Stag-Hunt mampu melanjutkan evolusi model untuk generasi yang jauh lebih banyak daripada NEAT (88 berbanding 47). Dengan kata lain, CATNeuro mampu menemukan peningkatan pada populasi dalam jangka waktu yang lebih lama daripada NEAT. Perhatikan bahwa kondisi penghentian adalah "tidak ada peningkatan yang terlihat dalam 50 generasi" untuk kedua mekanisme. Ini berarti bahwa proses dilanjutkan selama 50 generasi setelah tidak ada peningkatan yang tercatat pada model terbaik saat ini.

Sejujurnya, mungkin saja ukuran populasi yang lebih besar dapat menghasilkan hasil yang lebih baik untuk NEAT karena hal itu dapat memungkinkan NEAT untuk mempertahankan keragaman model yang lebih besar. Masalahnya adalah model DL membutuhkan waktu lama untuk dilatih, dan populasi yang lebih besar akan memperlambat proses yang sudah sangat panjang. CA Stag-Hunt menunjukkan pemanfaatan sumber daya yang lebih baik, dalam kondisi eksperimen, daripada NEAT sendiri. Sinergi antara kedua pendekatan tersebut jelas terlihat di sini.



Gambar 9.19 Distribusi jumlah generasi ketika model terbaik berada di bawah setiap percobaan.

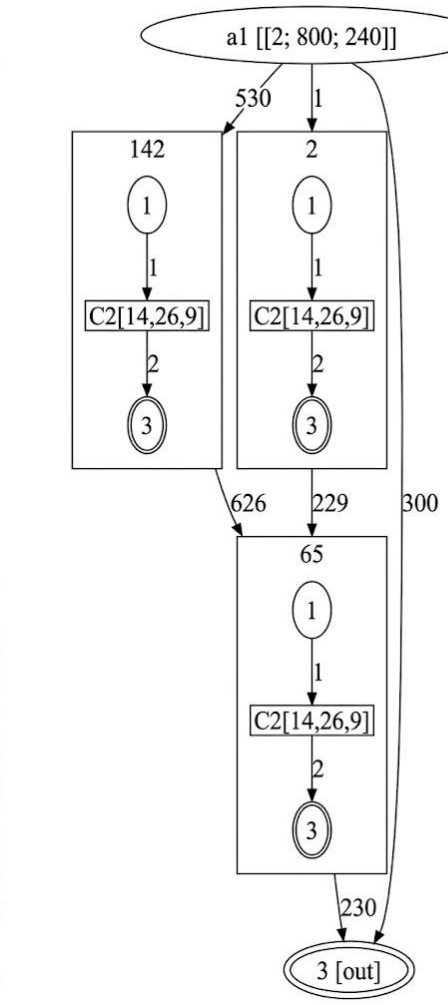
Model terbaik dari Stag-Hunt ditunjukkan pada Gambar 9.20. Ini menunjukkan dua lapisan konvolusi 2D yang terhubung secara seri (ditunjukkan sebagai "C2 [kernel, filter, stride]" dalam diagram) dengan koneksi "skip". Ini adalah struktur yang umum muncul untuk masalah ini. Strukturnya agak kontraintuitif — seorang ahli manusia tidak mungkin merancang jaringan seperti itu. Ini menunjukkan kegunaan NAS dalam mengidentifikasi struktur baru tetapi berharga. Kerugian di luar sampel pada 0.19 piksel persegi sangat rendah menunjukkan akurasi yang baik. Namun, jumlah parameter pada 783 K mungkin besar untuk inferensi waktu nyata pada perangkat tertanam. Oleh karena itu, penyetelan lebih lanjut mungkin diperlukan dalam hal pengkodean input alternatif, peningkatan sampel piksel, dan/atau kuantisasi ke 4 atau 8 bit untuk mengurangi kebutuhan komputasi untuk inferensi waktu nyata.



Gambar 9.20 Model pelacakan gerakan trailer terbaik: Stag-Hunt [Loss = 0.19; Parameter = 783 K].

Model terbaik yang dihasilkan NEAT ditunjukkan pada Gambar 9.21. Kerugian di luar sampel pada 0.19 piksel persegi sama dengan Stag-Hunt, namun, jumlah parameternya jauh lebih tinggi yaitu 920 K. Strukturnya menunjukkan lapisan konvolusi paralel dan serial bersama dengan koneksi "skip". Tidak jelas apakah substruktur konvolusi paralel menambah nilai untuk model ini. Pemangkasan lebih lanjut dapat dilakukan dengan bantuan ahli manusia untuk mendapatkan struktur model akhir. Namun, model terbaik yang dihasilkan Stag-Hunt jelas lebih sederhana dan sama efektifnya.

Secara keseluruhan, dengan mempertimbangkan model Stag-Hunt dan NEAT, sejumlah kecil "gaya" arsitektur muncul. Sebagian besar model yang berkinerja baik mengikuti satu gaya atau lainnya. Dengan demikian, NAS mengidentifikasi arsitektur yang menjanjikan yang dapat diperiksa dan disempurnakan lebih lanjut oleh ahli manusia, mengurangi waktu/upaya keseluruhan yang diperlukan untuk mengatasi masalah tertentu.



Gambar 9.21 Model pelacakan gerakan trailer terbaik: NEAT [Loss = 0:19; Parameter = 920K].

9.7 KESESUAIAN CA UNTUK SISTEM KOMPLEKS

CA sangat cocok untuk pemecahan masalah dalam sistem kompleks seperti yang dibuktikan oleh karya-karya sebelumnya. NAS adalah bidang baru yang penting yang membutuhkan solusi untuk menemukan struktur model DL optimal dan nilai hiperparameter masalah optimasi yang kompleks dan terstruktur secara hierarkis. CATNeuro adalah investigasi tentang penerapan CA untuk NAS. Ia mengambil konsep evolusi grafik dari neuroevolusi dan menggabungkannya dengan pendorong evolusi yang terinspirasi oleh sistem budaya. Secara konseptual, evolusi grafik menarik karena memungkinkan munculnya arsitektur baru dengan menerapkan batasan struktural a priori minimal.

Pertama, serangkaian struktur grafik solusi sederhana diusulkan. Ini secara progresif "dikomplekskan" dalam generasi berikutnya hingga kemajuan lebih lanjut terhenti. Pembelajaran sebelumnya menunjukkan bahwa neuroevolusi bekerja dengan baik dalam praktik. Namun, neuroevolusi menggunakan strategi evolusi sederhana yang dirancang sebelum munculnya DL. Strategi neuroevolusi dirancang untuk memaksimalkan produktivitas sejumlah percobaan karena struktur model dan bobot parameter dipelajari secara bersamaan tanpa menggunakan back-propagation.

Sebaliknya, NAS saat ini adalah proses dua langkah di mana struktur model diusulkan terlebih dahulu dan kemudian bobot parameter dipelajari melalui pelatihan offline. Waktu pencarian didominasi oleh pelatihan model. Dalam skenario NAS, pendekatan hiper-heuristik CATNeuro terbukti berkinerja lebih baik daripada strategi neuroevolusi yang lebih sederhana.

Dua versi CATNeuro diperkenalkan. Keduanya berbeda dalam hal bagaimana pengetahuan dihasilkan oleh penerapan KS pada model grafik dalam jaringan. Pendekatan "Kebijaksanaan Kerumunan" membuat KS bersaing untuk model grafik yang tertanam dalam jaringan populasi. Yang kedua, "Perburuan Rusa", memungkinkan KS untuk bersaing dan bekerja sama dalam menghasilkan pengetahuan baru dalam jaringan.

Pertama, kedua mekanisme tersebut dibandingkan pada tugas benchmark untuk mengembangkan pengontrol jaringan saraf untuk game pertarungan 2D berkecepatan tinggi. Data pelatihan untuk tugas ini diperoleh dengan bantuan pembelajaran penguatan (*reinforcement learning*). Stag-Hunt secara statistik berkinerja lebih baik dalam hal kerugian model dan kompleksitas daripada WTD untuk aplikasi waktu nyata tersebut.

Konfigurasi pemenang CATNeuro, Stag-Hunt, kemudian diuji coba terhadap NEAT pada masalah pelacakan gerakan trailer dunia nyata. CATNeuro ditemukan secara statistik lebih baik daripada NEAT untuk akurasi model dan kompleksitas model (dalam hal jumlah parameter). Lebih lanjut, distribusi kerugian dan ukuran model menunjukkan bahwa CATNeuro menghasilkan model yang lebih konsisten di berbagai percobaan eksperimen.

Dalam pekerjaan mendatang, rentang model yang dapat ditangani oleh CATNeuro direncanakan untuk diperluas untuk menangani jenis lapisan atau modul lain, misalnya LSTM, perhatian, dll. CATNeuro menghasilkan representasi model abstrak independen dari toolkit DL spesifik apa pun. Penerjemah diperlukan untuk mengubah model abstrak menjadi model konkret. Saat ini, penerjemah hanya ada untuk toolkit CNTK Microsoft. Upaya sedang dilakukan untuk membuat penerjemah PyTorch. Dukungan untuk TensorFlow juga direncanakan. Pekerjaan selanjutnya akan membandingkan CATNeuro dengan penawaran NAS pesaing untuk platform ini. CATNeuro masih terus berkembang. Tujuannya adalah untuk menjadikannya penawaran yang layak untuk NAS bagi toolkit DL yang dominan.

BAB 10
PENGENALAN EMOSI AUDIO BERBASIS FUSI MULTIMODAL

Artikel ini berfokus pada penerapan alat-alat teori informasi baru di bidang fusi informasi. Transformasi dan fusi fitur sangat penting untuk kinerja fusi informasi, namun, sebagian besar karya yang ada bergantung pada statistik orde kedua, yang hanya optimal untuk distribusi seperti Gaussian. Dalam artikel ini, integrasi teknik fusi informasi dan pemeriksaan komponen entropi kernel menyediakan alat teori informasi baru. Fusi fitur direalisasikan menggunakan deskriptor entropi informasi dan dioptimalkan dengan estimasi entropi. Strategi fusi informasi multimodal baru untuk pengenalan emosi audio berdasarkan pemeriksaan komponen entropi kernel (KECA) telah dipresentasikan. Efektivitas solusi yang diusulkan dievaluasi melalui eksperimen pada dua basis data emosi audiovisual. Hasil eksperimen menunjukkan bahwa solusi yang diusulkan mengungguli metode yang ada, terutama ketika dimensi ruang fitur dikurangi secara substansial. Metode yang diusulkan menawarkan pemeriksaan teoritis umum yang memberi kita pendekatan untuk mengimplementasikan teori informasi ke dalam pembelajaran multimedia.

10.1 DASAR FUSI INFORMASI MULTIMODAL

Fusi informasi multimodal digambarkan sebagai suatu proses yang mengintegrasikan serangkaian sumber informasi ganda, fitur yang diekstrak, dan keputusan perantara untuk mencapai hasil pemeriksaan yang lebih andal. Meningkatnya minat dalam pembelajaran multimodalitas disebabkan oleh potensi kemampuannya untuk menghilangkan keterbatasan kritis tertentu dari modalitas tunggal. Antarmuka manusia-komputer tradisional seperti mouse dan keyboard dianggap terlalu membatasi untuk interaksi alami. Banyak upaya telah difokuskan pada berbagai sensor non-intrusif sehingga manusia dapat melakukan aktivitas mereka dengan cara yang lebih alami tanpa merasakan kehadiran sensor-sensor ini. Hal ini menuntut penggunaan multimodalitas, terutama untuk sistem multimedia. Oleh karena itu, tujuan fusi informasi ganda adalah untuk memanfaatkan informasi komplementer dan meningkatkan akurasi kinerja pengenalan secara keseluruhan. Karena sensor yang berbeda dapat membawa informasi yang berlebihan, komplementer, atau bahkan bertentangan, memanfaatkan semua data mengarah pada sistem yang mampu memahami pengguna secara lebih menyeluruh daripada menggunakan modalitas tunggal.

10.2 ISU-ISU KRITIS FUSI INFORMASI

Tantangan Penggabungan Informasi

Keuntungan penggabungan informasi disertai dengan beberapa kendala dalam proses pemeriksaan. Alasan utamanya adalah karena karakteristik yang berbeda dari berbagai modalitas yang terlibat, karena modalitas yang berbeda pada dasarnya ditangkap dalam berbagai format dengan kecepatan yang berbeda. Oleh karena itu, untuk menyelesaikan tugas dengan lebih baik, proses penggabungan perlu mengatasi beberapa masalah kritis, misalnya,

sinkronisasi berbagai modalitas. Oleh karena itu, pemilihan teknik dan algoritma yang digunakan untuk penggabungan merupakan faktor kunci bagi kinerja dan akurasi sistem multimodal.

Masalah penting lainnya yang terkait dengan teknik multimodal adalah variasi data masukan. Berbagai modalitas biasanya memiliki tingkat kepercayaan dan keandalan yang berbeda dalam mencapai keputusan. Selain itu, yang perlu lebih diperhatikan adalah bahwa modalitas yang berbeda mungkin independen atau berkorelasi, karena baik independensi maupun korelasi sama-sama memberikan wawasan yang berharga dalam berbagai skenario. Untuk mengatasi kekhawatiran yang disebutkan di atas, investigasi telah dilakukan terhadap kinerja teknik penggabungan informasi.

Tingkat Fusi

Skema fusi informasi yang ada biasanya diklasifikasikan menjadi lima modul: fusi tingkat sensor, fusi tingkat fitur, fusi tingkat skor, fusi tingkat keputusan, dan fusi hibrida [4, 5].

Fusi Tingkat Sensor

Fusi tingkat sensor, atau fusi tingkat data, mengacu pada integrasi informasi mentah dari dua atau lebih sensor. Meskipun fusi pada tingkat sensor diharapkan dapat meningkatkan akurasi pengenalan, biasanya tidak dapat digunakan dalam fusi multimodal karena ketidaksesuaian data dari modalitas yang berbeda.

Fusi Tingkat Fitur

Fusi tingkat fitur adalah jenis solusi yang menggabungkan vektor fitur yang berbeda, yang diperoleh baik dengan modalitas yang berbeda atau dengan menerapkan algoritma ekstraksi fitur yang berbeda pada modalitas yang sama. Fusi pada tingkat fitur juga disebut sebagai fusi awal. Meskipun tingkat fitur mengandung informasi yang lebih kaya tentang data mentah, sulit untuk mencapai sinkronisasi waktu dan format yang sama, dan sulit untuk mendapatkan korelasi antara fitur heterogen. Lebih jauh lagi, penggabungan vektor fitur dapat menghasilkan vektor fitur dengan dimensi tinggi, yang sangat meningkatkan beban komputasi.

Fusi Tingkat Skor

Fusi tingkat skor, yang dikenal sebagai fusi tingkat menengah, mengintegrasikan skor pencocokan yang diberikan oleh beberapa modalitas. Keuntungannya meliputi implementasi yang sederhana dan skalabilitas.

Fusi Tingkat Keputusan

Fusi tingkat keputusan juga disebut sebagai fusi akhir yang mengacu pada kombinasi nilai kemungkinan atau skor probabilitas yang diperoleh dari modalitas tunggal terpisah untuk membuat keputusan gabungan. Fusi pada tingkat ini memungkinkan fleksibilitas yang luas dalam memilih pengklasifikasi individual. Namun, fusi pada tingkat keputusan memiliki kekurangan, termasuk hilangnya korelasi pada tingkat fitur dan proses pembelajaran yang melelahkan.

Fusi Hibrida

Fusi hibrida adalah metode berdasarkan strategi yang disebutkan di atas yang telah dikembangkan oleh banyak peneliti, karena pendekatan fusi hibrida dapat menggabungkan manfaat dari fusi awal dan akhir.

10.3 REDUKSI DIMENSI DAN FUSI FITUR BERBASIS KERNEL

Karena pengukuran heterogen dari fitur atau modalitas yang berbeda, data yang diekstrak seringkali tidak tepat dan tidak lengkap. Sebelum klasifikasi, penghapusan data yang berlebihan, pengintegrasian informasi pelengkap, dan pemrosesan vektor fitur sangat penting. Oleh karena itu, metode transformasi dan fusi fitur digunakan untuk membuat subset fitur baru dengan kombinasi fitur asli.

Strategi linier biasanya digunakan untuk membuang komponen redundan dan mengurangi dimensi data yang tinggi. Tujuannya adalah untuk mendapatkan diskriminasi yang lebih tinggi dari data berdimensi rendah dari data asli berdimensi tinggi. Pendekatan linier yang banyak digunakan meliputi pemeriksaan diskriminan linier (LDA), pemeriksaan komponen utama (PCA), pemeriksaan korelasi kanonik (CCA), pemeriksaan faktor lintas modal (CFA) dan sebagainya. Salah satu metode yang banyak digunakan adalah pemeriksaan korelasi kanonik (CCA). CCA adalah pendekatan statistik yang mewujudkan pengurangan dimensi linier dan fusi fitur dengan menghitung proyeksi linier yang berkorelasi maksimal. Tidak seperti CCA, pemeriksaan faktor lintas modal (CFA) adalah metode baru untuk asosiasi lintas model. Selain penghapusan noise, ia menyediakan kemampuan seleksi fitur yang dikurangi.

Metode-metode ini mengubah data berdimensi tinggi menjadi fitur berdimensi rendah yang sebagian besar bergantung pada statistik orde kedua, seperti varians, korelasi, kesalahan kuadrat rata-rata, dll. Ekstraksi fitur biasanya didasarkan pada nilai eigen teratas dan vektor eigen yang sesuai dari matriks tertentu. Misalnya, PCA menggunakan varians sebagai metrik. Namun, dasar dari metode yang ada, statistik orde kedua, hanya optimal untuk distribusi seperti Gaussian. Oleh karena itu, jika suatu distribusi sangat berbeda dari Gaussian, alat statistik orde kedua merupakan estimator yang buruk.

Selain itu, sebagian besar metode yang ada mengasumsikan bahwa selalu ada hubungan linier di antara data asli. Namun, dalam banyak situasi, ekstraksi fitur non-linier diperlukan. Untuk mencapai transformasi non-linier, metode kernel diusulkan sebagai implementasi non-linier, yang mengarah pada kernel PCA, kernel CCA dan kernel CFA. Namun, metode-metode ini masih memilih nilai eigen dan vektor eigen teratas dari matriks kernel, dan tidak mengungkapkan sifat dari kumpulan data masukan.

Untuk mengatasi masalah ini, kami termotivasi untuk menerapkan pemeriksaan komponen entropi kernel (KECA) sebagai alternatif. Tidak seperti metode yang ada yang bergantung pada statistik orde kedua dari kumpulan data, pemeriksaan komponen entropi kernel (KECA) didasarkan pada teori informasi dan mempertahankan entropi Renyi maksimum dari data masukan dengan jumlah fitur yang diekstrak terkecil. Dari aspek klasifikasi, KECA memungkinkan pemeriksaan data nonlinier dan menangkap statistik orde tinggi dari data. KECA tidak sesuai dengan nilai eigen dan vektor eigen teratas dari matriks kernel. Di sisi lain, transformasi dan fusi fitur berdasarkan KECA dicapai dengan kontribusi terbesar dari estimasi entropi. Pada bagian berikut, beberapa alat teori informasi diperkenalkan dan metode optimal secara teori informasi, pemeriksaan komponen entropi kernel, dijelaskan. Hasil eksperimen

menunjukkan bahwa kinerja strategi yang diusulkan lebih baik daripada metode yang ada, terutama ketika ruang fitur memiliki pengurangan dimensi yang substansial.

10.4 TEORI ENTROPI DAN METODE KERNEL

Entropi Shannon

Dalam fisika, entropi adalah cara estimasi yang berkorelasi dengan kuantitas keacakan kinematik. Namun dalam teori informasi, entropi bukan lagi konsep fisik dalam termodinamika. Konsep entropi informasi pertama kali diperkenalkan oleh Claude Shannon sebagai ukuran ketidakpastian statistik yang banyak digunakan dalam teori komunikasi. Konsep ini menyediakan kerangka kerja matematis untuk mengukur dan merumuskan sifat informasi di luar hukum fisika dan dengan cepat diterima oleh komunitas sains dan teknik. Shannon menyatakan bahwa ukuran jumlah informasi yang terkandung dalam serangkaian peristiwa dapat dinyatakan dengan entropi Shannon. Entropi Shannon ditunjukkan sebagai

$$H_s(X) = -\sum_k p(x_k) \log p(x_k) \tag{10.1}$$

Atau

$$H_s(X) = -\int f_x(x) \log f_x(x) \tag{10.2}$$

di mana $p(x_k)$ dan $f_x(x)$ masing-masing adalah fungsi kepadatan probabilitas diskrit dan kontinu dari himpunan data, dan k adalah jumlah total himpunan data dalam kasus diskrit.

Entropi Shannon diyakini sebagai metrik yang efektif untuk mengukur ketidakpastian kuantitas acak dalam hal perilaku probabilistik dari sumber informasi. Sifat fundamental entropi adalah skalar tunggalnya yang mengukur ketidakpastian dalam bentuk kepadatan probabilitas. Sifatnya yang kuat dan elegan membuatnya lebih cocok untuk menangkap karakteristik informasi. Selain itu, dapat diinterpretasikan sebagai cara untuk mengukur konten informasi. Ini juga dapat diperluas untuk mengukur ketidakmiripan antar data.

Entropi Renyi

Entropi Renyi adalah salah satu generalisasi entropi informasi yang banyak digunakan. Entropi Renyi orde 2 dari variabel acak X dinyatakan sebagai

$$H_\alpha(X) = \frac{1}{1-\alpha} \log\left(\sum_1^N p_k^\alpha\right) \tag{10.3}$$

Atau

$$H_\alpha(X) = \frac{1}{1-\alpha} \log\left(\int f_x^\alpha(x) dx\right) \tag{10.4}$$

di mana $\alpha \geq 1$. Entropi Shannon adalah kasus khusus dari entropi Renyi ketika 2 mendekati satu. Dalam artikel ini, entropi kuadrat Renyi digunakan, karena lebih fleksibel daripada entropi Shannon. Entropi kuadrat Renyi dinyatakan sebagai

$$H(X) = -\log\left(\int p^2(x)dx\right) \tag{10.5}$$

di mana $p(x)$ adalah fungsi kepadatan probabilitas yang dihasilkan oleh himpunan data $D = x_1, \dots, x_N$.

Alasan utama untuk memilih $\alpha = 2$ adalah bahwa nilai entropi dapat diperkirakan secara elegan oleh pdf $p(x)$ (fungsi kepadatan probabilitas) yang dihasilkan dari himpunan data D dan estimator entropi yang efisien secara komputasi dapat diwujudkan. Estimator entropi dicapai dengan mengganti pdf dengan estimator kepadatan non-parametrik yang disebut estimator kepadatan jendela Parzen.

Metode Kernel

Metode kernel banyak digunakan dalam masalah nonlinier pemeriksaan data. Prinsip dasar metode kernel adalah memetakan data asli ke ruang fitur dengan transformasi nonlinier dan menggunakan algoritma linier di ruang baru. Jika ruang input terdiri dari $x_i \in \mathbb{R}^{\mathbb{d}}$ dalam himpunan X, pemetaan nonlinier dinyatakan sebagai berikut.

$$\phi: \mathbb{R}^{\mathbb{d}} \to F \tag{10.6}$$

$$x \to \phi(x) \tag{10.7}$$

di mana $F \in \mathbb{R}^{\mathbb{l}}, l \geq d$.

Fungsi kernel adalah fungsi k yang memenuhi kondisi berikut yang dikenal sebagai trik kernel.

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle, \quad x_i, x_j \in X \tag{10.8}$$

Ekspresi eksplisit dari pemetaan non-linier ϕ sulit ditentukan. Namun, trik kernel memungkinkan kita untuk mendapatkan hasil perkalian dalam ruang fitur dengan dimensi yang mungkin tak terbatas secara langsung tanpa menghitung pemetaan eksplisit ϕ. Ini memungkinkan kita untuk menyelesaikan masalah non-linier dengan beroperasi di ruang fitur berdimensi tinggi dengan algoritma pembelajaran mesin linier yang diekspresikan melalui hasil perkalian dalam. Namun, fungsi kernel harus memenuhi kondisi Mercer, yaitu semi-positif definit.

Beberapa fungsi kernel yang banyak digunakan meliputi kernel linier $k(x_i, x_j) = \langle x_i, x_j \rangle$, kernel polinomial $k(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d$, kernel eksponensial $k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{2\sigma^2}\right)$ dan kernel Gaussian. Kernel Gaussian didefinisikan sebagai berikut:

$$k_\sigma(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{10.9}$$

di mana σ dikenal sebagai ukuran kernel.

Estimator Kepadatan Jendela Parzen

Estimator kepadatan non-parametrik adalah strategi yang memperkirakan distribusi probabilitas dari suatu himpunan data tanpa asumsi bentuk atau parameter. Salah satu estimator non-parametrik adalah jendela Parzen yang dapat dilihat sebagai fungsi kernel dan menciptakan hubungan erat antara teori informasi dan metode kernel. Estimator kepadatan jendela Parzen diberikan oleh

$$\tilde{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{10.10}$$

di mana $K(\cdot)$ adalah kernel dan h adalah parameter penghalus yang disebut lebar. Estimator densitas non-parametrik diperoleh dengan mengganti pdf sebenarnya dengan estimator jendela Parzen-nya. Oleh karena itu, dengan menggunakan metode jendela Parzen yang dikenal konsisten dan efisien, estimator non-parametrik untuk entropi tidak memerlukan estimasi eksplisit dari pdf.

10.5 TRANSFORMASI FITUR BERDASARKAN PEMERIKSAAN KOMPONEN ENTROPI KERNEL

Dalam pemeriksaan komponen entropi kernel, ukuran entropi adalah entropi kuadrat Renyi dan estimasi densitas direalisasikan oleh estimator densitas jendela Parzen. Entropi kuadrat Renyi kontinu dinyatakan sebagai

$$H(p) = -\log\left(\int p^2(x)dx\right) = -\log V(p) \tag{10.11}$$

di mana $V(p) = \int p^2(x)dx = E\{p(x)\}$. $V(p)$ dapat dilihat sebagai ekspektasi terhadap kerapatan $p(x)$.

Karena sifat monoton fungsi logaritma, kita hanya perlu mempertimbangkan kuantitas $V(p) = \int p^2(x)dx$. Untuk memperkirakan $p(x)$ atau $V(p)$, digunakan estimator kerapatan jendela Parzen. Estimator kerapatan jendela Parzen dapat dilihat sebagai semacam realisasi trik kernel, dan dapat dianggap sebagai jumlah hasil perkalian dalam yang dihitung di ruang fitur. Kemudian estimator kerapatan jendela Parzen berdasarkan notasi kernel dapat ditulis ulang sebagai berikut.

$$\tilde{p}(x) = \frac{1}{N\sigma}\sum_{x_i \in D} K\left(\frac{x - x_i}{\sigma}\right) = \frac{1}{N}\sum_{x_i \in D} k_\sigma(x, x_i) \tag{10.12}$$

di mana $k_\sigma(x, x_i)$ adalah kernel dari estimator kepadatan jendela Parzen yang berpusat di x_i dan σ menunjukkan ukuran kernel.

Kami mengasumsikan kernel Parzen semi-positif definit dengan fungsi Gaussian $k_\sigma(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$. Teorema konvolusi untuk fungsi Gaussian menyatakan bahwa konvolusi dari dua fungsi Gaussian adalah fungsi Gaussian lain, dengan $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$. Karena $V(p)$ dapat dianggap sebagai $E\{p(x)\}$, kita dapat memperoleh estimasi berikut.

$$\tilde{V}(p) = \frac{1}{N}\sum_{i=1}^{N} \tilde{p}(x_i) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k_\sigma(x_i, x_j) = \frac{1}{N^2} 1^T K 1 \tag{10.13}$$

di mana elemen (i, j) dari matriks kernel $N \times N$ K sama dengan $k(x_i, x_j)$, dan 1 adalah vektor $N \times 1$ yang berisi semua angka satu. Oleh karena itu, entropi kuadrat Renyi dapat diekspresikan secara ringkas dalam bentuk matriks kernel.

Selanjutnya, kita dapat mengekspresikan estimasi entropi Renyi dalam bentuk nilai eigen dan vektor eigen dari matriks kernel melalui dekomposisi eigen. Dekomposisi eigen dari K dapat ditunjukkan sebagai $K = EDE^T$, di mana D adalah matriks diagonal yang berisi nilai eigen $\lambda_1, \ldots, \lambda_N$ dan E adalah matriks dengan vektor eigen yang sesuai $\alpha_1, \ldots, \alpha_N$ sebagai kolom. Oleh karena itu, estimasi entropi Renyi empiris sama dengan elemen-elemen dari matriks kernel yang sesuai. Ekspresi di atas dapat ditulis ulang untuk mendapatkan hasil berikut.

$$\tilde{V}(p) = \frac{1}{N^2} 1^T K 1 = \frac{1}{N^2} 1^T E D E^T 1 = \frac{1}{N^2}\sum_{i=1}^{N} \left(\sqrt{\lambda_i}\alpha_i^T 1\right)^2 \tag{10.14}$$

di mana λ_i dan α_i adalah nilai eigen dan vektor eigen ke-i dari matriks kernel jendela Parzen K, dan 1 adalah vektor $N \times 1$ yang berisi angka satu.

Ekspresi ini disebut nilai entropi. Dari ekspresi di atas, karena setiap suku $\sqrt{\lambda_i}\alpha_i^T$ berkontribusi pada estimasi entropi total, mudah diamati bahwa baik nilai eigen maupun vektor eigen berkontribusi pada estimasi entropi. Di sisi lain, nilai eigen tertentu dan vektor eigen yang sesuai memberikan kontribusi lebih besar pada estimasi entropi total. Nilai eigen dan vektor eigen dipilih berdasarkan estimasi entropi terbesar, bukan nilai eigen terbesar, yang menghasilkan hasil yang berbeda dari metode yang ada, seperti kernel PCA, kernel CCA, dll. Oleh karena itu, pemeriksaan komponen entropi kernel adalah teknik transformasi fitur yang memproyeksikan ruang asli ke subruang fitur yang direntang oleh sumbu utama kernel yang sesuai dengan kontribusi terbesar dari entropi Renyi.

Prosedur algoritma pemeriksaan komponen entropi kernel diringkas sebagai berikut. Pertama-tama, data input adalah vektor fitur $X = \{x_1, x_2, \dots, x_N\}$, yang membutuhkan transformasi dan fusi fitur. Seperti yang disebutkan di atas, kita memilih fungsi Gaussian sebagai fungsi kernel dan memperoleh matriks kernel K dengan elemen $K_{ij} = k(x_i, x_j)$. Kemudian dekomposisi eigen dari K dilakukan. Langkah selanjutnya adalah memilih estimasi entropi terbesar n pertama berdasarkan $\sqrt{\lambda_i}\alpha_i^T$; Setelah itu himpunan data ruang fitur kernel $\phi_{eca} = D^{\frac{1}{2}}E^T$ dapat dihitung, karena kita dapat menyimpulkan bahwa $\phi_{eca}^T\phi_{eca} = \left(D^{\frac{1}{2}}E^T\right)^T D^{\frac{1}{2}}E^T = \boldsymbol{EDE^T} = \boldsymbol{K}$; Terakhir kita dapat menyelesaikan transformasi fitur dengan ϕ_{eca}.

Dari sudut pandang teori informasi, KECA mempertahankan sebanyak mungkin entropi Renyi antara ruang input dan subruang fitur kernel dengan jumlah fitur terkecil, sehingga isi informasi dari data input dan data yang ditransformasikan semaksimal mungkin serupa. Di sisi lain, dari sudut pandang fusi informasi, KECA membantu mewujudkan metode fusi semi-supervised yang dapat mengurangi dimensi fitur dengan menghilangkan redundansi data dan memanfaatkan komplementaritas data dalam bentuk ukuran entropi.

10.6 PENGENALAN EMOSI AUDIO BERDASARKAN KOMPONEN ENTROPI KERNEL

Pengenalan Emosi Audio

Karena informasi ucapan merupakan metode interaksi yang paling alami antara manusia dan mesin, studi tentang pengenalan emosi berdasarkan informasi audio telah menarik perhatian yang semakin meningkat akhir-akhir ini. Pengenalan emosi audio didefinisikan sebagai ekstraksi keadaan emosional dari informasi ucapan pembicara. Banyak solusi telah diusulkan untuk memproses ucapan lisan dan mengidentifikasi informasi emosional. Namun, tugas ini membutuhkan kecerdasan buatan yang cukup untuk mengenali sinyal suara manusia. Terlepas dari kemajuan yang cukup besar dalam pemrosesan ucapan, kita masih jauh dari membangun sistem interaksi emosi ucapan alami yang sukses antara manusia dan mesin.

Pengenalan emosi berdasarkan informasi audio berguna untuk aplikasi komunikasi alami dan ramah antara manusia dan komputer, seperti interaksi manusia-komputer cerdas, keamanan dan pengawasan, hiburan dan pendidikan online, dll. Beberapa aplikasi spesifik baru-baru ini termasuk robot rumah tangga cerdas untuk interaksi alami dan ramah dengan manusia dan sistem pengenalan emosi tipe ketakutan yang didedikasikan untuk pengawasan visual-audio.

Pengenalan emosi berdasarkan informasi ucapan merupakan tantangan karena aspek-aspek berikut. Pertama-tama, tidak begitu jelas fitur ucapan mana yang paling cocok untuk mengkarakterisasi berbagai keadaan emosional secara efisien tanpa bergantung pada konten leksikal atau pembicara. Pemilihan fitur yang tepat dan efisien diyakini akan memengaruhi kinerja pengenalan secara signifikan. Masalah kedua ditambahkan oleh variabilitas akustik, yang dipengaruhi oleh gaya bicara dan kecepatan bicara dari berbagai pembicara. Fitur ucapan yang umum digunakan seperti nada dan energi rentan terhadap kekurangan ini.

Fitur Audio untuk Pengenalan Emosi

Karena data ucapan tidak selalu stasioner, sinyal ucapan biasanya dibagi menjadi frame dalam pemrosesan ucapan. Sinyal dianggap stasioner dalam setiap frame. Fitur ucapan yang banyak digunakan meliputi fitur ucapan kontinu dan fitur ucapan spektral. Fitur ucapan kontinu (atau fitur ucapan prosodik) seperti nada dan energi juga disebut fitur lokal dan diekstrak dari setiap frame sinyal audio, sedangkan fitur ucapan spektral juga disebut fitur global dan dihitung sebagai statistik dari semua fitur ucapan.

Fitur ucapan kontinu telah banyak digunakan dalam pengenalan emosi berdasarkan sinyal audio. Dipercaya bahwa fitur kontinu seperti nada dan energi merupakan indikator utama keadaan emosi pembicara dan menyampaikan banyak informasi emosional. Karena informasi temporal yang ada dalam informasi ucapan, fitur ucapan kontinu lebih unggul dalam hal akurasi klasifikasi. Dengan menggunakan sejumlah vektor fitur kontinu yang cukup, pengklasifikasi kompleks seperti *support vector machine* (SVM) dan *hidden Markov model* (HMM) dapat dilatih dengan andal dan parameter model dapat diestimasi secara akurat.

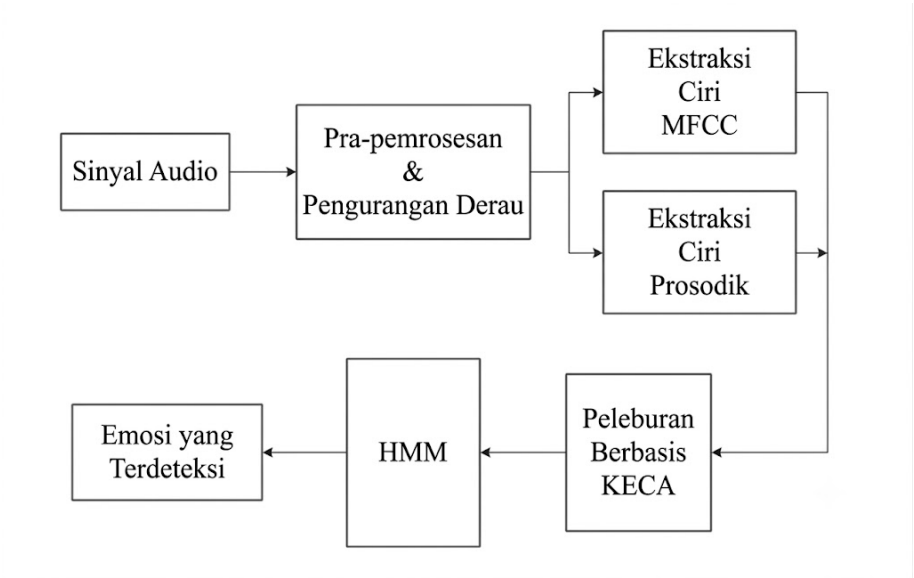
Selain fitur kontinu yang bergantung pada waktu, fitur spektral dipilih sebagai representasi berbeda untuk informasi ucapan. Telah ditunjukkan bahwa fitur spektral berdasarkan pemeriksaan cepstral seperti MFCC (koefisien cepstral frekuensi Mel) dan LPCC (koefisien cepstral prediktif linier) jelas mengungguli fitur berbasis linier seperti LPC (koefisien prediktor linier). Fitur spektral memiliki representasi yang berbeda dari sifat informasi. Selain itu, jumlah fitur ucapan spektral lebih sedikit, sehingga algoritma pemilihan fitur dieksekusi lebih cepat, dan efisiensi pelatihan pengklasifikasi relatif tinggi.

Belum ada kesepakatan mengenai fitur mana yang lebih efisien dan sesuai untuk pengenalan emosi, tetapi banyak peneliti mengklaim bahwa fitur lokal dan global efisien dalam membedakan berbagai keadaan emosi. Karena fitur ucapan kontinu dan fitur ucapan spektral memiliki kelebihan dan keterbatasan masing-masing, diyakini bahwa integrasi fitur kontinu dan spektral menyampaikan lebih banyak informasi tentang keadaan emosi manusia. Jika satu modalitas gagal mendeteksi emosi, modalitas lain dapat membantu meningkatkan kinerja. Oleh karena itu, hubungan komplementer dari modalitas ini menghasilkan akurasi klasifikasi yang lebih tinggi daripada yang dicapai oleh fitur tunggal.

Penggabungan Informasi Fitur Audio Berdasarkan KECA

Terdapat banyak pembelajaran tentang keadaan emosi yang mengungkapkan bahwa setidaknya enam emosi dasar bersifat universal, oleh karena itu artikel ini berfokus pada enam emosi utama termasuk kebahagiaan, kejutan, kesedihan, ketakutan, jijik, dan kemarahan. Banyak solusi telah diperkenalkan untuk pengenalan emosi berdasarkan informasi audio. Kinerja pengenalan emosi berdasarkan penggabungan audio telah ditunjukkan oleh beberapa karya. Namun, hal itu masih jauh dari solusi utama karena akurasi dan efisiensi solusi yang diusulkan masih kurang memuaskan. Selain itu, sebagian besar karya memperlakukan fitur audio sebagai informasi independen dan belum membangun hubungan yang erat di antara keduanya. Dalam artikel ini, solusi penggabungan bimodal baru untuk sinyal audio pada tingkat fitur telah diteliti. Gambar 10.1 menggambarkan diagram blok dari strategi yang diusulkan.

Pertama-tama, prosedur pemilihan dan fusi fitur harus dilakukan untuk mengekstrak fitur-fitur penting dan mengurangi dimensi ruang fitur. Untuk mengurangi noise, metode ambang koefisien wavelet diimplementasikan pada tahap pra-pemrosesan. Karena tepi depan dan belakang tidak memiliki pesan yang berguna, maka tepi tersebut dihilangkan. Karena pemeriksaan audio dapat diandalkan ketika sinyal stasioner, pemeriksaan waktu singkat dapat dilakukan dalam interval waktu singkat stabilitas artikulasi. Oleh karena itu, sinyal audio harus di-windowing menjadi serangkaian urutan yang juga disebut frame. Sebelum ekstraksi fitur, jendela Hamming berukuran 512 titik dengan tumpang tindih 50% antara jendela yang berdekatan dikalikan pada setiap frame ucapan.



Gambar 10.1 Diagram blok fusi tingkat fitur untuk pengenalan emosi audio berdasarkan KECA.

Fitur prosodik berkaitan dengan konten ritmis sinyal audio. Fitur ini biasanya direpresentasikan oleh statistik intensitas, pitch, frekuensi fundamental, frekuensi formant, dll. Dalam artikel ini, statistik dan variasi pitch, energi, panjang jeda, kecepatan bicara, dan frekuensi formant diekstrak sebagai fitur kontinu. Mengenai fitur spektral, MFCC (*Mel-frequency cepstral coefficient*) adalah salah satu alat pemeriksaan yang paling efektif di bidang pengenalan suara. Alat ini banyak digunakan karena secara fisik terkait dengan perilaku telinga manusia melalui pemeriksaan cepstral. Dalam artikel ini, fitur MFCC dihitung berdasarkan frame suara. Tiga belas koefisien pertama digunakan sebagai fitur yang berguna, karena sebagian besar energi informasi tersimpan dalam beberapa koefisien pertama. Statistik setiap koefisien, seperti median, rata-rata, deviasi standar, nilai maksimum, dan nilai minimum, digabungkan untuk membentuk vektor fitur. Setelah pemeriksaan kontinu dan spektral, vektor fitur audio diperoleh melalui fitur gabungan dari frame berurutan.

Sebelum klasifikasi, pengurangan dimensi dan fusi fitur harus dilakukan, karena kinerja klasifikasi sebagian besar bergantung pada kemampuan diskriminatif dari fitur yang ditransformasikan. Vektor fitur yang besar mengandung informasi yang kaya tentang

modalitas, tetapi biasanya mengalami kutukan dimensi. Dalam masalah ini, titik data dari vektor fitur yang besar menjadi jarang, sehingga himpunan data sampel yang terbatas mungkin tidak cukup mewakili distribusi yang mendasarinya untuk klasifikasi. Selain keterbatasan ini, pengklasifikasi yang kompleks tidak bekerja dengan baik jika fitur input memiliki dimensi yang sangat tinggi.

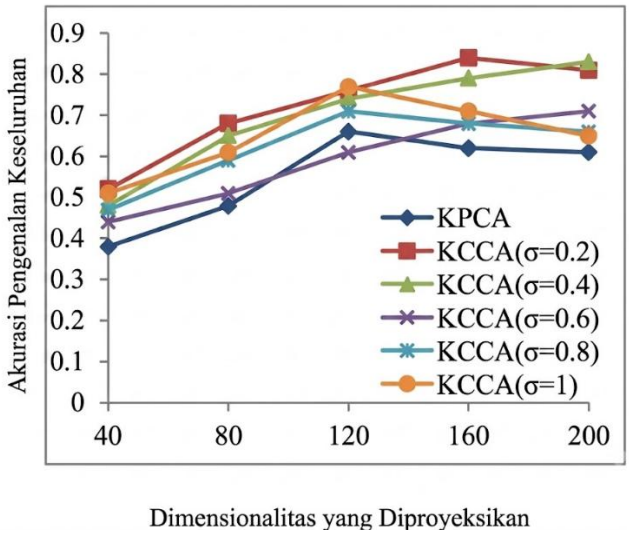
Untuk mengurangi masalah ini, fitur audio yang diekstrak diperiksa menggunakan pendekatan fusi yang diusulkan berdasarkan pemeriksaan komponen entropi kernel (KECA) dalam artikel ini. Strategi fusi fitur mencapai dua jenis peningkatan. Yang pertama adalah mengintegrasikan semua fitur audio dengan mempertahankan konten informasi secara maksimal dan memilih subset fitur yang mempertahankan karakteristik fitur asli. Yang kedua adalah mengubah ruang asli menjadi ruang yang ditransformasikan. Pengurangan dimensi dicapai dengan menghasilkan vektor fitur baru berdasarkan domain yang ditransformasikan.

Selama tahap klasifikasi, fitur yang dihasilkan dilihat sebagai input ke HMM (model Markov tersembunyi) dengan campuran kepadatan Gaussian. HMM dilatih sebagai pengklasifikasi karena kinerjanya yang luar biasa dalam memodelkan karakteristik temporal sinyal audio. Output dari HMM adalah kemungkinan urutan data tertentu yang dapat dianggap sebagai keadaan emosi yang terdeteksi.

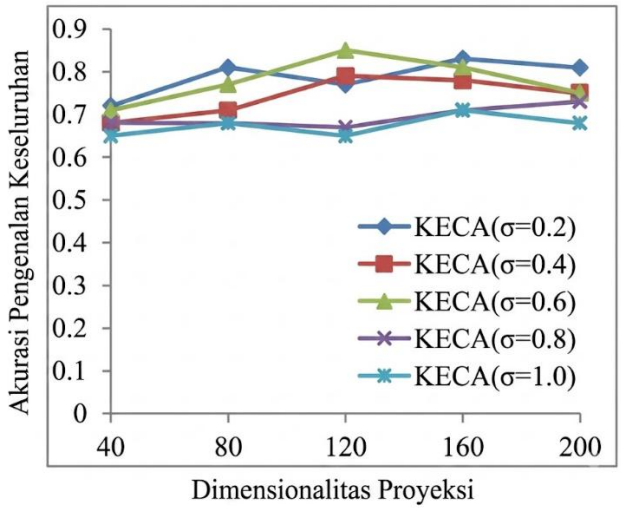
10.7 HASIL DAN EVALUASI EKSPERIMEN SISTEM PENGENALAN EMOSI

Untuk mengevaluasi efektivitas strategi yang diusulkan, eksperimen ekstensif telah dilakukan pada basis data emosi RML dan basis data emosi eNTERFACE. Basis data emosi RML berisi 720 sampel ekspresi emosi audiovisual. Terdapat delapan subjek manusia yang berbicara dalam enam bahasa. Enam emosi dasar manusia diekspresikan: marah, jijik, takut, sedih, terkejut, dan bahagia. Sampel direkam pada laju sampling 22.050 Hz menggunakan digitalisasi 16-bit saluran tunggal dengan laju bingkai 30 fps. Basis data emosi eNTERFACE berisi 44 subjek yang berasal dari 14 negara berbeda dan mengekspresikan 6 emosi dasar, dengan laju sampling 48.000 Hz dan laju bingkai 25 fps. Dalam eksperimen, setiap sampel audio telah dipotong menjadi 2 detik dan dibagi menjadi 10 segmen. Dimensi fitur audio secara empiris ditetapkan menjadi 240. Prosedur evaluasi didasarkan pada validasi silang. Untuk pelatihan, 75% sampel dipilih secara acak sedangkan sisanya untuk pengujian.

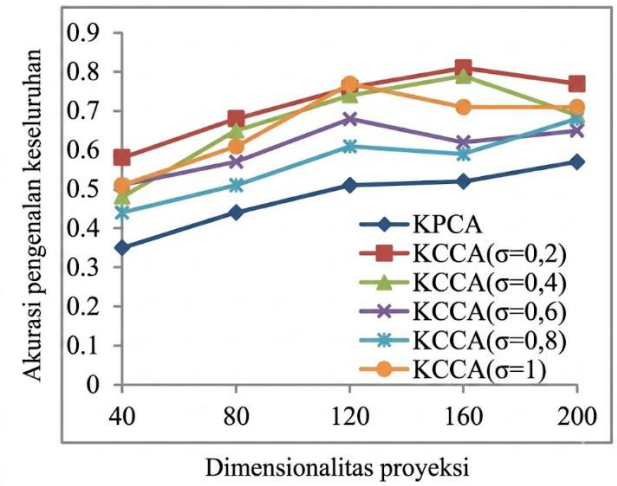
Gambar 10.2 menggambarkan perbandingan akurasi pengenalan keseluruhan antara KPCA dan KCCA pada basis data RML dan Gambar 10.3 menunjukkan hasil eksperimen basis data RML berdasarkan KECA.



Gambar 10.2 Hasil eksperimen basis data emosi RML berdasarkan KPCA dan KCCA.



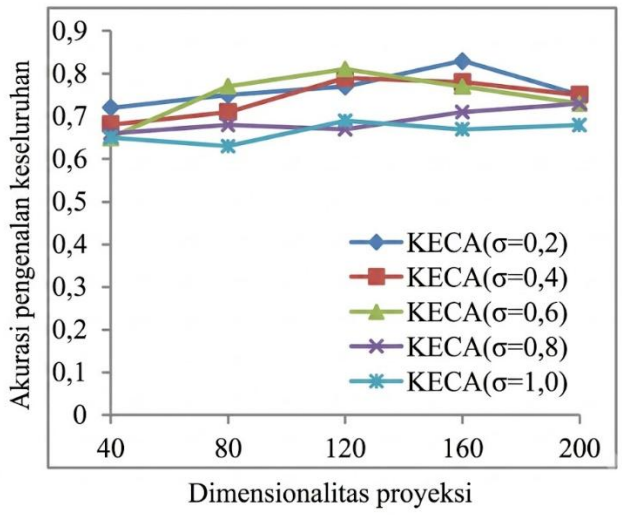
Gambar 10.3 Hasil eksperimen basis data emosi RML berdasarkan KECA.



Gambar 10.4 Hasil eksperimen basis data emosi eNTERFACE berdasarkan KPCA dan KCCA.

Gambar 10.4 menunjukkan kinerja keseluruhan KPCA dan KCCA pada basis data eNTERFACE dan Gambar 10.5 menampilkan kinerja KECA pada basis data eNTERFACE. σ menunjukkan ukuran kernel di semua gambar. Dari gambar-gambar ini, dapat dilihat bahwa KECA memiliki akurasi yang lebih baik daripada KPCA dan KCCA, dan mengungguli KPCA dan KCCA dalam pengurangan dimensi dan kinerja akurasi. Akurasi pengenalan keseluruhan dari metode yang diusulkan berdasarkan KECA stabil bahkan jika dimensi proyeksi rendah. Ini menunjukkan bahwa fitur yang diekstrak pada dimensi tinggi mungkin mengandung data yang berlebihan atau bising.

Setelah diproses dengan metode fusi berdasarkan KECA, sebagian besar informasi yang berguna dipertahankan dan akurasi yang stabil tercapai. Di sisi lain, kinerja KCCA dan KPCA sangat menurun jika dimensi proyeksi berkurang, yang berarti bahwa metode fusi tidak dipilih dengan tepat dan hasil yang menurun dihasilkan. Hasil ini menunjukkan peningkatan yang signifikan dalam pengurangan dimensi dan kemampuan untuk mempertahankan informasi yang berguna dari solusi yang diusulkan.



Gambar 10.5 Hasil eksperimen basis data emosi eNTERFACE berdasarkan KECA.

Emosi yang Diinginkan	Emosi Terdeteksi					
	Kebahagiaan	**Jijik**	**Takut**	**Marah**	**Terkejut**	**Kesedihan**
Kebahagiaan	**79.58**	4.10	4.55	3.78	5.75	2.24
Jijik	6.41	**69.38**	5.89	6.42	5.57	6.33
Takut	6.64	5.14	**71.85**	4.78	6.83	4.76
Marah	4.78	2.54	2.56	**83.45**	4.32	2.35
Terkejut	6.34	3.58	3.95	6.13	**75.14**	4.86
Kesedihan	7.69	6.39	5.29	6.53	5.43	**68.67**

Gambar 10.6 Matriks kebingungan kinerja rata-rata pada dua basis data berdasarkan KECA.

Gambar 10.6 adalah matriks kebingungan kinerja rata-rata berdasarkan KECA. Gambar 10.7 menunjukkan matriks kebingungan kinerja rata-rata berdasarkan KPCA. Gambar 10.8 menunjukkan hasil matriks kebingungan kinerja rata-rata berdasarkan KCCA. Ketiga gambar ini menampilkan matriks kebingungan kinerja rata-rata pada basis data emosi RML dan basis data emosi eNTERFACE. Dari gambar-gambar ini, jelas terlihat bahwa hasilnya cukup memuaskan dan menggembirakan. Dibandingkan dengan metode berdasarkan KPCA dan KCCA, strategi berdasarkan KECA memiliki akurasi dan stabilitas yang lebih baik.

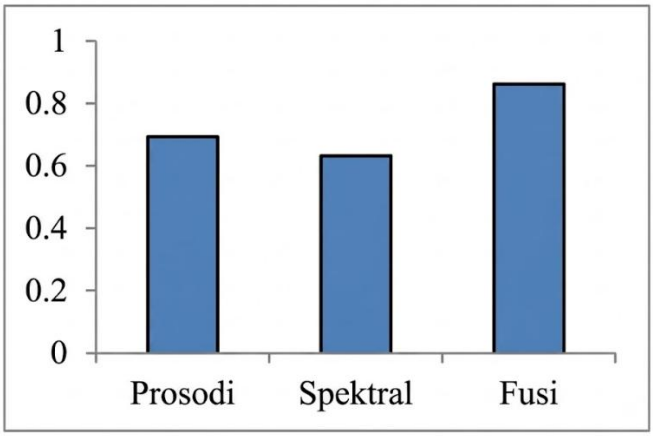
Gambar 10.9 dan 10.10 menunjukkan perbandingan akurasi pengenalan rata-rata antara metode fusi dan non-fusi pada basis data emosi RML dan basis data emosi eNTERFACE. Gambar 10.11 dan 10.12 menampilkan matriks kebingungan hasil fusi dan non-fusi. Teramati bahwa hasil fusi berdasarkan KECA lebih unggul daripada yang diperoleh oleh metode non-fusi dengan modalitas tunggal yang terlibat. Ini menegaskan efektivitas solusi fusi berdasarkan pemeriksaan komponen entropi kernel.

Emosi yang Diinginkan	**Emosi Terdeteksi**					
	Kebahagiaan	**Jijik**	**Takut**	**Marah**	**Terkejut**	**Kesedihan**
Kebahagiaan	**71.34**	8.53	2.60	5.35	4.85	7.33
Jijik	10.67	**57.74**	5.66	9.54	11.34	5.05
Takut	8.44	7.17	**61.91**	7.93	11.34	3.21
Marah	3.18	8.04	7.51	**72.94**	4.69	3.64
Terkejut	5.34	4.05	5.81	4.75	**68.63**	11.42
Kesedihan	13.42	2.21	4.20	11.42	9.31	**59.44**

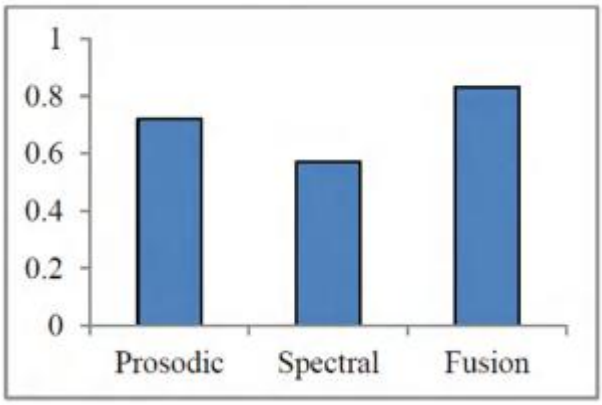
Gambar 10.7 Matriks kebingungan kinerja rata-rata pada dua basis data berdasarkan KPCA.

Emosi yang Diinginkan	**Emosi Terdeteksi**					
	Kebahagiaan	**Jijik**	**Takut**	**Marah**	**Terkejut**	**Kesedihan**
Kebahagiaan	**72.52**	4.34	6.77	3.91	3.25	9.21
Jijik	10.29	**60.85**	4.12	8.98	6.42	9.34
Takut	5.89	6.32	**63.54**	7.31	11.84	5.10
Marah	6.10	9.01	3.98	**69.11**	7.21	4.59
Terkejut	11.85	2.67	3.12	10.53	**69.23**	2.60
Kesedihan	10.60	5.53	7.43	7.02	8.29	**61.13**

Gambar 10.8 Matriks kebingungan kinerja rata-rata pada dua basis data berdasarkan KCCA.



Gambar 10.9 Perbandingan antara hasil fusi dan hasil non-fusi pada basis data emosi RML.



Gambar 10.10 Perbandingan antara hasil fusi dan hasil non-fusi pada basis data emosi eNTERFACE.

Emosi yang Diinginkan	Emosi Terdeteksi					
	Kebahagiaan	**Jijik**	**Takut**	**Marah**	**Terkejut**	**Kesedihan**
Kebahagiaan	**82.41**	1.23	5.82	2.33	3.01	5.20
Jijik	6.21	**73.89**	5.09	4.77	5.63	4.41
Takut	7.32	4.74	**74.52**	3.41	6.99	3.02
Marah	2.13	2.38	2.88	**85.89**	3.40	3.32
Terkejut	5.74	4.22	6.12	4.21	**76.01**	3.70
Kesedihan	6.40	6.91	3.28	5.90	3.17	**74.34**

Gambar 10.11 Matriks kebingungan hasil fusi berdasarkan dua basis data emosi.

Emosi yang Diinginkan	Emosi Terdeteksi					
	Kebahagiaan	**Jijik**	**Takut**	**Marah**	**Terkejut**	**Kesedihan**
Kebahagiaan	**73.12**	0.98	7.06	6.74	9.54	2.56
Jijik	10.71	**70.78**	3.67	9.03	2.40	3.41
Takut	9.23	3.97	**65.21**	3.51	9.21	8.87

Marah	4.31	1.29	7.22	**73.90**	4.81	8.47
Terkejut	3.90	11.95	5.73	4.83	**67.45**	6.14
Kesedihan	7.41	4.67	8.02	9.20	8.21	**62.49**

Gambar 10.12 Matriks kebingungan hasil non-fusi berdasarkan dua basis data emosi.

Singkatnya, dari hasil eksperimen di atas, disimpulkan bahwa kinerja keseluruhan KECA mengungguli metode yang ada, seperti KPCA dan KCCA. Selain itu, KECA memberikan akurasi pengenalan terbaik dan pengurangan dimensi yang lebih efisien, serta hasil pengenalan yang lebih stabil.

10.8 KESIMPULAN KONTRIBUSI KECA

Artikel ini bertujuan untuk membahas isu-isu fusi informasi multimodal dalam aplikasi multimedia. Beberapa isu kritis, termasuk tantangan fusi, tingkat fusi, dan kekurangan metode fusi yang ada, disajikan. Sebuah alat teoretis informasi baru, pemeriksaan komponen entropi kernel (KECA), telah dijelaskan. Dalam artikel ini, KECA diterapkan pada aplikasi pengenalan emosi audio. Solusi baru fusi tingkat fitur untuk pengenalan emosi berdasarkan pemeriksaan komponen entropi kernel (KECA) telah diperkenalkan. Metode yang disajikan dalam artikel ini menunjukkan kemampuan yang baik dalam pengurangan dimensi tanpa banyak memengaruhi kinerja akurasi dalam aplikasi multimedia. Hasil eksperimen yang ekstensif menunjukkan kelayakan strategi yang diusulkan.

BAB 11
DETEKSI PERISTIWA VIDEO BERBASIS ENSEMBLE DEEP LEARNING

Dengan ledakan data multimedia, deteksi peristiwa semantik dari video telah menjadi topik yang menuntut dan menantang. Selain itu, ketika data memiliki distribusi data yang miring, deteksi peristiwa yang menarik juga perlu mengatasi masalah ketidakseimbangan data. Perkembangan pembelajaran mendalam baru-baru ini telah menjadikannya bagian penting dari banyak sistem Kecerdasan Buatan (AI). Hingga saat ini, berbagai arsitektur pembelajaran mendalam telah diusulkan untuk berbagai aplikasi seperti Pemrosesan Bahasa Alami (NLP) dan pemrosesan gambar. Meskipun demikian, masih tidak praktis bagi satu model untuk bekerja dengan baik untuk berbagai aplikasi. Oleh karena itu, dalam artikel ini, diusulkan kerangka kerja pembelajaran mendalam ensemble baru yang dapat digunakan dalam berbagai skenario dan dataset. Kerangka kerja yang diusulkan mampu menangani masalah overfitting serta kehilangan informasi yang disebabkan oleh model tunggal. Selain itu, ia mengurangi masalah data yang tidak seimbang dalam data multimedia dunia nyata. Seluruh kerangka kerja mencakup serangkaian ekstraktor fitur pembelajaran mendalam yang terintegrasi dengan algoritma ensemble yang ditingkatkan berdasarkan metrik kinerja untuk data yang tidak seimbang. Pengklasifikasi *Support Vector Machine* (SVM) digunakan sebagai lapisan terakhir dari setiap komponen pembelajaran mendalam dan juga sebagai pembelajar lemah dalam modul ensemble. Kerangka kerja dievaluasi pada dua dataset video skala besar dan tidak seimbang (yaitu, bencana dan TRECVID). Hasil eksperimen yang ekstensif menggambarkan keunggulan dan efektivitas kerangka kerja yang diusulkan. Ini juga menunjukkan bahwa kerangka kerja yang diusulkan mengungguli beberapa metode pembelajaran mendalam yang terkenal, serta fitur konvensional yang terintegrasi dengan pengklasifikasi yang berbeda.

11.1 LATAR BELAKANG DETEKSI PERISTIWA VIDEO

Data multimedia merupakan sumber informasi yang unik yang menghadirkan peluang dan tantangan. Selama beberapa tahun terakhir, sumber multimedia (misalnya, Flickr, Twitter, YouTube, dll.) telah menghasilkan sejumlah besar data. Misalnya, pengguna aktif Twitter menghasilkan sekitar 500 juta tweet setiap hari atau pengguna YouTube mengunggah ratusan jam video setiap menit. Ledakan data multimedia ini, termasuk gambar, audio, video, dan teks, telah mengubahnya menjadi masalah big data.

Kebutuhan akan pemeriksaan semantik otomatis dalam data multimedia tampak jelas dalam banyak aplikasi dunia nyata. Secara khusus, deteksi peristiwa video merupakan tugas penting dan menantang dalam sistem manajemen multimedia. Selama dekade terakhir, para peneliti telah mencari teknik otomatis untuk mendeteksi peristiwa dan konsep yang paling menarik dari video. Beberapa aplikasi deteksi peristiwa semantik video meliputi deteksi kejahatan, pengambilan data bencana alam, dan pengenalan peristiwa menarik dari pertandingan olahraga.

Terlepas dari semua peluang yang diberikan oleh big data multimedia dan pemeriksaan semantik, beberapa tantangan perlu diatasi. Tantangan pertama adalah bagaimana mengelola data yang besar dan multimodal tersebut secara efektif dan efisien. Selain itu, masalah data tidak seimbang (data dengan distribusi tidak seragam atau miring) rumit dan teknik konvensional yang sederhana tidak mampu menanganinya. Masalah terakhir ini telah banyak diamati dalam aplikasi deteksi konsep multimedia. Deteksi kanker dari data medis (misalnya, gambar MRI) atau deteksi penipuan dari transaksi bank adalah contoh data tidak seimbang.

Selain itu, pengambilan video yang berisi bencana alam di antara ribuan video YouTube juga dapat dianggap sebagai masalah data tidak seimbang. Dalam contoh terakhir di mana metadata dan informasi tekstual mungkin tidak selalu dapat diandalkan dan akurat, kita mencari peristiwa yang menarik atau kecil (video yang berisi informasi bencana); sementara distribusi kelas (kecil dan besar) tidak seragam. Distribusi data yang miring dalam tugas deteksi peristiwa video tidak dapat dihindari dan teknik pembelajaran konvensional sebagian besar bias terhadap kelas besar, sementara kita tertarik pada kelas kecil. Meskipun masalah ini telah dipelajari dalam literatur selama bertahun-tahun, namun masih menjadi topik hangat terutama pada dataset multimodal skala besar.

Pembelajaran mendalam adalah topik yang sedang berkembang yang telah menghasilkan banyak terobosan untuk pembelajaran multimedia termasuk pemrosesan gambar, penambangan teks, dan pengenalan ucapan. Hal ini telah menarik banyak perhatian di industri dan akademisi dalam beberapa tahun terakhir. Secara umum, arsitektur grafik mendalam berisi kaskade lapisan, yang terdiri dari beberapa transformasi linier dan non-linier. Dengan menggunakan arsitektur ini, model pembelajaran mendalam melakukan abstraksi tingkat tinggi dari data mentah. Namun, masih terbatas pembelajaran yang menangani masalah data yang tidak seimbang dalam data multimedia menggunakan pembelajaran mendalam.

Sampai saat ini, banyak arsitektur pembelajaran mendalam telah diusulkan untuk berbagai aplikasi. Namun, hampir tidak mungkin bagi satu model untuk bekerja dengan baik untuk semua skenario dan dataset. Sulit juga untuk menangani data multimedia yang tidak seimbang dan besar karena overfitting, kehilangan informasi, dan bias tambahan. Oleh karena itu, fusi multi-kelas dapat dimanfaatkan untuk meningkatkan teknik pembelajaran mendalam. Dalam studi ini, kami mengusulkan kerangka kerja pembelajaran mendalam ensemble, yang tidak hanya mengatasi masalah data yang tidak seimbang dalam big data multimedia, tetapi juga mengurangi kehilangan informasi dan masalah overfitting yang disebabkan oleh model tunggal. Terinspirasi oleh keberhasilan besar pembelajaran mendalam, kerangka kerja ini digunakan untuk pemeriksaan fitur mendalam dengan aplikasi pada deteksi peristiwa video. Selanjutnya, pendekatan ensemble dikembangkan berdasarkan kinerja setiap pembelajar lemah (pengklasifikasi *Support Vector Machine* (SVM)) pada setiap set fitur mendalam untuk meningkatkan deteksi peristiwa semantik dalam dataset yang tidak seimbang.

11.2 DETEKSI PERISTIWA VIDEO

Penyebaran dan ledakan data multimedia termasuk audio, gambar, video, dan teks telah menyebabkan munculnya berbagai aplikasi dalam sistem manajemen multimedia. Di antaranya, pemeriksaan video adalah salah satu topik yang paling menantang. Lebih khusus lagi, deteksi peristiwa video otomatis telah menarik banyak perhatian dalam aplikasi dunia nyata seperti pengawasan video, manajemen informasi bencana, deteksi kejahatan, dan lain sebagainya. Namun demikian, untuk memenuhi semua persyaratan dalam aplikasi pemeriksaan video, ada beberapa tantangan yang perlu diatasi. Pertama-tama, keberadaan konsep atau peristiwa semantik yang menarik dalam data video bisa jadi jarang. Dengan kata lain, kita menghadapi masalah data yang tidak seimbang. Fenomena ini telah banyak terlihat dalam aplikasi dunia nyata, seperti pengenalan aktivitas dan penambangan video.

Secara umum, solusi untuk data yang tidak seimbang dapat dibagi menjadi tiga kategori. Kelompok yang paling umum adalah metode resampling seperti oversampling dan undersampling. Teknik-teknik ini terutama menyesuaikan distribusi data untuk menciptakan dataset yang seimbang. Kelompok kedua adalah teknik sensitif biaya yang memberikan penalti pada instance yang salah klasifikasi dan mencoba untuk meningkatkan kinerja kelas minoritas. Kelompok terakhir mencakup metode berbasis kernel dan pembelajaran aktif. Tujuan utama kelompok ini adalah untuk memanfaatkan algoritma klasifikasi yang lebih kuat untuk menangani data yang tidak seimbang secara alami. Di antara mereka, pembelajaran ensemble (dapat diklasifikasikan sebagai metode sensitif biaya) telah menunjukkan keberhasilan yang signifikan dalam beberapa tahun terakhir.

Dalam tugas multimedia, fitur itu penting. Oleh karena itu, bagaimana mengekstrak fitur yang berguna dari data untuk meningkatkan hasil deteksi akhir adalah tantangan substansial lainnya dalam sistem multimedia. Pertama, ruang fitur dikurangi dengan menghitung korelasi antara pasangan fitur dan fitur-fitur dengan korelasi rendah terhadap fitur lainnya dihilangkan. Setelah itu, teknik yang disebut *Hidden Coherent Feature Groups* (HCFGs) digunakan untuk pengelompokan fitur. Akhirnya, untuk setiap kelompok fitur, sebuah pengklasifikasi dilatih dan skornya digabungkan dengan pengklasifikasi lain untuk deteksi kejadian akhir. Informasi spasial dan temporal dari urutan video diintegrasikan untuk menghasilkan representasi fitur baru. Dalam karya tersebut, penulis menerapkan medan aliran optik dan detektor sudut Harris3D yang terintegrasi dengan teknik penguatan ensemble yang didasarkan pada dua algoritma klasifikasi: representasi sparse dan pengklasifikasi jarak Hamming.

Deep learning merupakan kemajuan besar dalam pembelajaran mesin dan penambangan data. Meskipun deep learning memiliki sejarah panjang dalam AI, ia telah menjadi solusi untuk banyak masalah dalam beberapa tahun terakhir dengan penggunaan daya komputasi yang besar dan desain algoritma yang efisien. *Convolutional Neural Networks* (CNN) adalah salah satu algoritma yang paling efektif dalam deep learning. Algoritma ini dikembangkan sekitar tahun 1990 yang terinspirasi oleh pembelajaran Hubel dan Wiesel tentang korteks visual hewan. CNN telah banyak diterapkan dalam pengolahan gambar dan meningkatkan jaringan saraf feed-forward tradisional.

Perbedaan utamanya terletak pada konektivitas lokal dan topologi pembagian bobot yang secara signifikan membatasi kompleksitas jaringan. Berbeda dengan jaringan saraf tradisional yang sangat rentan terhadap overfitting dan sulit diinterpretasikan (kotak hitam), algoritma deep learning baru lebih mudah diinterpretasikan karena pemodelan lokal yang kuat dari arsitektur deep learning. Selain itu, tidak hanya daya komputasi yang meningkat pesat selama dekade terakhir, tetapi juga ide-ide baru, algoritma, dan arsitektur jaringan dalam beberapa tahun terakhir telah menyebabkan revolusi besar dalam aplikasi multimedia, terutama dalam pengenalan gambar dan deteksi objek. Alexnet adalah salah satu arsitektur pembelajaran mendalam baru pertama yang secara signifikan meningkatkan kinerja jaringan saraf dalam dan mengklasifikasikan dataset yang sangat besar (misalnya, ImageNet di ILSVRC 2012) ke dalam seribu kelas.

Kemudian, arsitektur CNN mendalam baru yang disebut GoogleNet diusulkan dalam yang dievaluasi secara eksperimental pada tugas klasifikasi dan deteksi ILSVRC 2014. Jaringan ini diperluas baik dalam kedalaman maupun lebar dibandingkan dengan jaringan konvensional sementara biaya komputasi tetap konstan. Jaringan sukses lainnya disebut *Regions with convolutional Neural Networks* (R-CNN). Ini mencakup dua submasalah: deteksi objek dan klasifikasi gambar. Akhirnya, Microsoft memperkenalkan jaringan deep learning Residual pada tahun 2015 yang mengalahkan manusia dalam deteksi objek di ILSVRC 2015. Setelah itu, tidak ada peningkatan signifikan dalam arsitektur deep learning yang dilaporkan kecuali yang mencoba mengintegrasikan keunggulan teknik yang ada saat ini dengan cara yang wajar.

Dalam artikel ini, kami tidak hanya mengekstrak parameter yang bermanfaat dari jaringan deep learning yang disebutkan di atas, tetapi juga mengembangkan teknik ensemble baru dan canggih untuk mengintegrasikan hasilnya secara efektif. Motivasinya adalah fakta bahwa pengklasifikasi tunggal mungkin tidak selalu mampu menangani dataset besar dan bising dengan distribusi miring. Namun, algoritma ensemble dapat digunakan untuk meningkatkan kinerja klasifikasi dengan memanfaatkan beberapa pengklasifikasi. Sebuah pengklasifikasi ensemble yang terukur diusulkan berdasarkan keputusan beberapa "penilai" yang sebelumnya dilatih dan dievaluasi pada pengklasifikasi dan fitur yang berbeda. Dalam karya lain, sebuah algoritma ensemble yang ditingkatkan secara positif diusulkan. Algoritma ini menangani masalah data yang tidak seimbang dalam pengambilan peristiwa video dengan menggabungkan teknik berbasis sampling dengan pendekatan klasifikasi berbasis fusi.

Metode ini diterapkan pada dataset video olahraga yang tidak seimbang untuk mendeteksi peristiwa yang menarik. Teknik ensemble juga diterapkan pada beberapa jaringan saraf tradisional untuk mengurangi efek overfitting. Misalnya, jaringan saraf ensemble disajikan dalam. Pendekatan ini dikembangkan menggunakan pendekatan sampling bootstrap yang terintegrasi dengan jaringan saraf konvensional untuk mendeteksi peristiwa langka dalam video sepak bola.

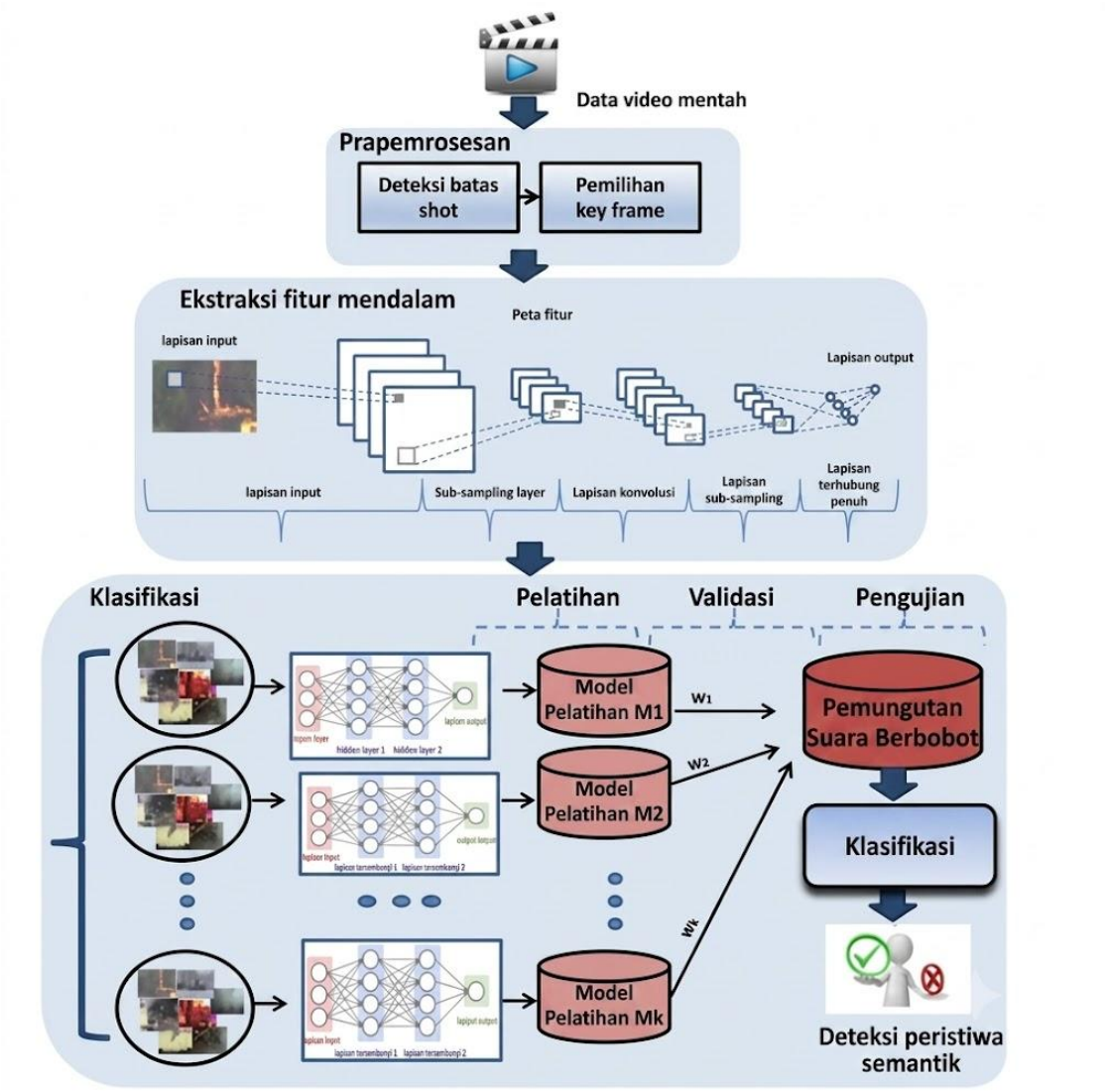
11.3 KERANGKA KERJA ENSEMBLE DEEP LEARNING

Kerangka kerja *Ensemble Deep Learning* (EDL) terdiri dari campuran ekstraktor fitur yang menggunakan teknik deep learning yang terintegrasi dengan algoritma ensemble yang

diusulkan. Seluruh kerangka kerja dibagi menjadi tiga modul utama, yaitu (1) pra-pemrosesan, (2) ekstraksi fitur mendalam, dan (3) klasifikasi (seperti yang ditunjukkan pada Gambar 11.1). Modul klasifikasi juga mencakup langkah-langkah pelatihan, validasi, dan pengujian.

Praproses

Langkah pertama dalam setiap pemeriksaan data adalah bagaimana melakukan praproses data yang tidak terstruktur. Langkah ini spesifik domain dan setiap jenis data (misalnya, audio, gambar, video, dan teks) mungkin memerlukan rutinitas praprosesnya sendiri. Untuk pemrosesan video dalam pembelajaran ini, kami menggunakan pendekatan deteksi batas bidikan otomatis dan tanpa pengawasan berdasarkan teknik pelacakan objek dan segmentasi gambar. Dengan menggunakan pendekatan ini, batas bidikan dari setiap video mentah dideteksi dan bingkai pertama dari setiap bidikan dipilih sebagai bingkai kunci. Bingkai pertama dari setiap bidikan dapat dianggap sebagai yang paling khas karena merupakan batas dari dua bidikan yang berurutan. Setelah praproses, bingkai kunci yang dipilih digunakan untuk deteksi peristiwa dalam video.



Gambar 11.1 Kerangka kerja pembelajaran mendalam ensemble yang diusulkan.

Ekstraksi Fitur Mendalam

Sebelum tahun 2010, pembelajaran dalam visi komputer sebagian besar berfokus pada peningkatan fitur buatan tangan dan menghasilkan atribut yang lebih diskriminatif dari data. Beberapa fitur buatan tangan yang umum dan ampuh meliputi HOG, CEDD, dan SIFT untuk data visual dan MFCC untuk data aural. Namun, kemajuan ini mulai melambat antara tahun 2010 dan 2012 dengan munculnya teknik pembelajaran mendalam baru seperti CNN. Dalam beberapa tahun terakhir, pembelajaran mendalam berkembang sangat pesat dan telah meningkatkan hasil kinerja secara signifikan. Dalam studi ini, kami juga memutuskan untuk memanfaatkan algoritma yang sedang berkembang ini dan menerapkannya sebagai ekstraktor fitur pada data kami. Untuk tujuan ini, beberapa model ekstraksi fitur yang kaya dan mendalam diintegrasikan dengan cara yang tepat. Modul ekstraksi fitur mendalam didasarkan pada algoritma CNN dan memanfaatkan model pra-terlatih menggunakan pembelajaran transfer.

Jaringan Saraf Konvolusional

CNN adalah versi lanjutan dari jaringan *MultiLayer Perceptron* (MLP). Namun, pada CNN, sebagian besar neuron terhubung secara lokal dan bukan terhubung sepenuhnya, yang sangat meningkatkan kecepatan pelatihan dan mengurangi overfitting dengan menghilangkan sejumlah besar parameter dalam jaringan. Tidak seperti MLP, input setiap lapisan dalam CNN diatur dalam tiga dimensi: lebar, tinggi, dan kedalaman. Misalnya, untuk input gambar 256 × 256 × 3, lebar dan tinggi sama dengan 256 dan 3 adalah kedalaman input ini yang mengacu pada nomor saluran (misalnya, RGB). Setiap neuron dalam CNN terhubung ke wilayah kecil dari lapisan sebelumnya. Secara keseluruhan, ada tiga lapisan utama untuk membangun arsitektur jaringan konvolusional: (1) Lapisan konvolusional, (2) Lapisan pooling, dan (3) Lapisan terhubung penuh. Sebuah CNN mencakup tumpukan lapisan konvolusional yang diikuti oleh lapisan pooling dan biasanya diakhiri dengan lapisan terhubung penuh seperti yang ditunjukkan pada modul ekstraksi fitur mendalam pada Gambar 11.1.

Pada lapisan konvolusional, neuron terhubung ke wilayah lokal dalam input, masing-masing menghasilkan produk titik antara wilayah kecil dalam volume input dan bobot yang sesuai. Akibatnya, sejumlah peta fitur dihasilkan dengan mengkonvolusikan (menggeser) filter di atas semua lokasi spasial dalam data masukan. Dengan kata lain, peta fitur diperoleh dengan konvolusi data masukan dengan filter linier (dan penambahan suku bias) diikuti oleh fungsi aktivasi nonlinier seperti yang diilustrasikan pada Persamaan (11.1), di mana x_{ij}^{k} mengacu pada peta fitur k^{th} pada lapisan tertentu, i dan j adalah dimensi masukan, dan x_{ij}^{k-1} adalah data masukan dari lapisan sebelumnya. Filter lapisan k^{th} ditentukan oleh W_{ij}^{k} (bobot) dan b_{j}^{k} (bias). Terakhir, fungsi aktivasi atau nonlinieritas ditunjukkan dengan f. Salah satu fungsi aktivasi yang paling banyak digunakan untuk pembelajaran mendalam adalah *Rectified Linear Unit* (ReLu) $(f(x) = max(0, x))$ yang meningkatkan nonlinieritas dan menunjukkan kinerja yang lebih baik dibandingkan dengan fungsi konvensional (misalnya, sigmoid, tanh, dll.).

$$x_{ij}^{k} = f\left(\left(W_{ij}^{k} * x_{ij}^{k-1}\right) + b_{j}^{k}\right) \tag{11.1}$$

Setelah setiap lapisan konvolusi, terdapat lapisan pooling yang menerapkan operasi downsampling nonlinier sepanjang lebar dan tinggi (dimensi spasial) dari input gambar yang diberikan pada Persamaan (11.2), di mana β_{ij}^{k} adalah bias perkalian dan $down(.)$ adalah fungsi subsampling (misalnya, max, average, dll.). Oleh karena itu, dengan menggunakan lapisan pooling, ukuran setiap peta aktivasi dikurangi, yang membuat representasi lebih mudah dikelola. Ini juga menangani overfitting dan memberikan kekokohan tambahan pada jaringan.

$$x_{ij}^{k} = f\left(\beta_{ij}^{k} down\left(x_{ij}^{k-1}\right) + b_{j}^{k}\right) \tag{11.2}$$

Terakhir, lapisan yang terhubung sepenuhnya digunakan sebagai lapisan terakhir CNN untuk menghitung penalaran tingkat tinggi atau skor kelas. Mirip dengan jaringan saraf tradisional,

semua neuron atau peta aktivasi dari lapisan subsampling konvolusional sebelumnya terhubung sepenuhnya ke satu neuron di lapisan ini.

Ekstraksi Fitur Menggunakan Pembelajaran Transfer

Dalam artikel ini, beberapa arsitektur pembelajaran mendalam yang canggih dan sukses digunakan untuk ekstraksi fitur visual. Untuk tujuan ini, alih-alih melatih seluruh CNN dari awal, kami mengambil model referensi yang telah dilatih sebelumnya dan memperlakukan jaringan konvolusional sebagai ekstraktor fitur untuk dataset baru. Model referensi ini telah dilatih sebelumnya pada dataset skala sangat besar. Secara khusus, kami memilih model-model yang memiliki dampak lebih besar pada bidang pengolahan gambar dalam beberapa tahun terakhir. Dataset ImageNet yang berisi jutaan gambar dengan 1000 kategori konsep digunakan untuk melatih semua model tersebut. Oleh karena itu, kami menjalankan model-model yang telah dilatih sebelumnya ini pada dataset kami dan menghasilkan fitur (juga dikenal sebagai kode CNN) untuk semua gambar. Kumpulan fitur ini selanjutnya digunakan untuk klasifikasi. Tabel 11.1 menyajikan ringkasan model CNN yang digunakan dalam artikel ini untuk ekstraksi fitur.

Tabel 11.1 Model referensi pra-terlatih untuk ekstraksi fitur.

Metode	Tantangan	# Lapisan	# Kategori	Dataset
AlexNet	ILSVRC 2012	8	1000	ImageNet
R-CNN	ILSVRC 2013	7	200	ImageNet
	VOC 2012			PASCAL VOC
GoogleNet	ILSVRC 2014	22	1000	ImageNet
ResNet	ILSVRC 2015	152	1000	ImageNet
	COCO 2015			COCO

Seperti yang dapat dilihat dari tabel, berbagai model dengan jumlah lapisan dan arsitektur yang berbeda digunakan dan dijelaskan lebih lanjut di bawah ini:

- AlexNet dapat dianggap sebagai upaya pertama dan yang paling berpengaruh yang meningkatkan algoritma CNN untuk pemrosesan gambar dan membuatnya populer kembali. Pada dasarnya, jaringan ini berisi delapan lapisan: lima lapisan konvolusi diikuti oleh lapisan max pooling dan dropout (untuk mengatasi masalah overfitting), dan tiga lapisan terhubung penuh. Lapisan terhubung penuh terakhir menghasilkan 1000 kategori yang mungkin. Ia juga menggunakan augmentasi data, termasuk refleksi horizontal, translasi gambar, dan ekstraksi patch. AlexNet digunakan untuk memenangkan ILSVRC 2012 dan secara signifikan mengungguli runner-up kedua (lebih dari 10%) dalam hal tingkat kesalahan uji 5 teratas.
- CaffeNet dikembangkan oleh *Berkeley Vision and Learning Center* (BVLC). Ini adalah model referensi yang digunakan dalam perangkat lunak pembelajaran mendalam Caffe yang merupakan replikasi AlexNet dengan beberapa peningkatan. Misalnya, augmentasi data pencahayaan ulang tidak digunakan untuk melatih CaffeNet dan normalisasi diterapkan setelah lapisan pooling. Mirip dengan AlexNet, model ini dilatih pada dataset ImageNet.

- CNN berbasis wilayah (R-CNN) adalah jaringan berpengaruh lainnya dalam visi komputer, yang terutama dikembangkan untuk tugas deteksi objek. Prosesnya dapat dibagi menjadi dua bagian: pertama-tama menghasilkan proposal wilayah menggunakan teknik segmentasi bounding box dan kemudian menerapkan pengklasifikasi berbasis CNN untuk mendeteksi objek di lokasi tersebut. Dengan menggunakan teknik ini, kinerja meningkat secara signifikan (lebih dari 30%) dibandingkan dengan hasil terbaik pada PASCAL VOC 2012. Ia juga dilatih pada ImageNet di ILSVRC 2013 di mana ia menghasilkan 200 kategori.
- GooglNet adalah pemenang ILSVRC 2014 dalam dua tugas deteksi dan klasifikasi objek di mana ia memperkenalkan jaringan yang lebih dalam dan lebih luas. Ini dapat dianggap sebagai upaya pertama yang tidak hanya menumpuk beberapa lapisan konvolusi dan pooling secara berurutan, tetapi memiliki bagian-bagian jaringan yang terjadi secara paralel. Secara total, ia berisi 22 lapisan jaringan yang dalam yang secara luar biasa mempertimbangkan penggunaan memori dan daya dengan memanfaatkan sparsitas ekstra dari lapisan. Bagian utama dari jaringan ini dikenal sebagai "*Inception*" yang menghasilkan lokalitas yang lebih optimal dan mengulanginya secara spasial.
- Jaringan residual (ResNet) diperkenalkan oleh Microsoft di ILSVRC 2015 dengan kinerja yang hebat (tingkat kesalahan 3,6%). Selain itu, jaringan ini memenangkan tugas deteksi dan segmentasi COCO pada tahun 2015. Jaringan ini terdiri dari 152 lapisan (sangat dalam) dan memanfaatkan koneksi residual di seluruh jaringan. Tujuan utama jaringan ini adalah untuk mengatasi masalah degradasi: jaringan yang lebih dalam, akurasi yang lebih jenuh. Oleh karena itu, jaringan residual yang dalam diusulkan yang memungkinkan lapisan jaringan untuk menyesuaikan pemetaan residual.

Klasifikasi

Setelah mengekstrak fitur dari model referensi yang disebutkan di atas menggunakan pembelajaran transfer tanpa pengawasan, kami menggunakan teknik ensemble baru untuk mengurangi masalah overfitting dan meningkatkan kinerja. Pertama, fitur-fitur dalam yang diekstrak diperiksa untuk menemukan pentingnya setiap set fitur yang diekstrak dari setiap model pembelajaran mendalam secara terawasi. Selain itu, metode ensemble yang ditingkatkan diusulkan untuk mengoptimalkan Mengintegrasikan model-model yang telah dilatih. Metode ini secara efektif menyesuaikan koefisien bobot untuk modul klasifikasi (silakan lihat Gambar 11.1). Pertama, kita melatih k model klasifikasi (pengklasifikasi lemah dalam ensemble), masing-masing dilatih pada satu set fitur. Setelah itu, koefisien bobot setiap pengklasifikasi disesuaikan berdasarkan kinerja klasifikasinya pada dataset validasi. Langkah klasifikasi mencakup dua bagian: pembelajaran ensemble mendalam dan pengujian.

Pembelajaran Ensemble Mendalam

Prosedur pelatihan dari pembelajaran ensemble mendalam yang diusulkan diilustrasikan pada Algoritma 1. Pada langkah pertama, kita membagi dataset menjadi tiga bagian: pelatihan T, validasi V, dan pengujian T'. Misalkan set pelatihan didefinisikan sebagai $T = \{(t_1, c_1), (t_2, c_2), \dots, (t_N, c_N)\}$, di mana t_i adalah instance pelatihan i^{th}, c_i adalah kelas instance (misalnya, untuk tugas klasifikasi biner $c_i \in \{0,1\}$, dan N adalah ukuran set pelatihan.

Selain itu, kita menyimpan semua set fitur. diekstrak dari semua model pembelajaran mendalam di Fr. Ini adalah input lain dari algoritma pelatihan.

Pembelajaran ensemble yang diusulkan dapat dilihat sebagai agregasi bootstrap (*bagging*) yang melibatkan semua pembelajar lemah (pengklasifikasi) dalam pemungutan suara. Namun, dalam algoritma ini, pemungutan suara berbobot dihasilkan daripada memberikan bobot yang sama kepada setiap pembelajar. Selain itu, bobot diberikan berdasarkan metrik untuk data yang tidak seimbang.

Algoritma 1 Pelatihan *Ensemble Deep Learning*

Input: Data latih $T\{(t_i, c_i), i = 1,2, \dots, N\}$, Data validasi $V\{(v_i, c_i), i = 1,2, \dots, N_2\}$, Kumpulan fitur $Fr = \{F_j, j = 1,2, \dots, k\}$.

Output: Matriks bobot W_j, Model terlatih M_j.

1: **untuk setiap** $\boldsymbol{F_j} \in \boldsymbol{Fr(j = 1, \dots, k)}$ **lakukan**

2: $M_j \leftarrow \text{SVM}(T, F_j)$;

3: **akhir untuk**

4: **untuk setiap** $\boldsymbol{F_j} \in \boldsymbol{Fr(j = 1, \dots, k)}$ **lakukan**

5: $F1_j \leftarrow \text{VALIDATE}(V, F_j)$;

6: $W_j = \frac{F1_j}{\sum_{j=1}^{k} F1_j}$;

7: **akhir untuk**

8: **kembalikan** $\boldsymbol{W_j, M_j}$

Oleh karena itu, hasilnya ditingkatkan ke arah kelas minoritas, sementara kinerja kelas mayoritas dipertahankan setinggi mungkin. Pembelajar lemah atau model didefinisikan sebagai $M = \{M_j, j = 1,2, \dots, k\}$, masing-masing dilatih menggunakan SVM linier seperti yang ditunjukkan pada Baris 1–3 dari Algoritma 1, di mana k adalah jumlah total pembelajar lemah. SVM digunakan sebagai pengklasifikasi utama karena telah menunjukkan hasil yang menjanjikan ketika diintegrasikan dengan pembelajaran mendalam. Setelah pembelajar lemah $M_j(j = 1,2, \dots, k)$ dilatih Dengan menggunakan instance pelatihan, setiap model dievaluasi menggunakan set validasi V seperti yang ditunjukkan pada Baris 4–7 dari Algoritma 1. Untuk evaluasi dan penyesuaian koefisien bobot, ukuran F1 (Persamaan (11.3)) digunakan yang merupakan angka antara 0 (kasus terburuk) dan 1 (kasus terbaik).

$$F1 = 2 * \frac{Precision * Recall}{(Precision + Recall)} \tag{11.3}$$

di mana Presisi adalah rasio dari jumlah instance positif yang benar-benar diprediksi (TP) terhadap semua instance positif yang diprediksi (TP+FP); sedangkan Recall adalah rasio dari TP terhadap semua instance positif yang sebenarnya (TP+FN).

Selanjutnya, bobot setiap model terlatih M_j dihitung menggunakan rasio skor F1 yang sesuai $(F1_j)$ terhadap jumlah skor untuk semua model seperti yang ditunjukkan pada Persamaan (11.4).

$$W_j = \frac{F1_j}{\sum_{j=1}^{k} F1_j} \tag{11.4}$$

Koefisien bobot memberikan nilai yang lebih tinggi (probabilitas) kepada model yang lebih yakin dengan prediksinya. Terakhir, algoritma mengembalikan setiap model dan bobot W_j yang sesuai untuk digunakan lebih lanjut dalam modul pengujian.

Pengujian

Algoritma 2 mengilustrasikan prosedur pengujian. Masukan pertama dari algoritma ini mencakup himpunan pengujian $T'\{t_i', i = 1,2, \dots, N_3\}$, di mana t_i' adalah instance pengujian i^{th} dan N_3 adalah jumlah total instance pengujian. Selain instance, untuk setiap model pelatihan M_j, himpunan fitur F_j dan matriks bobot W_j diberikan sebagai masukan dari algoritma ini untuk memprediksi label dari instance pengujian. Untuk mencapai hal ini, kita menghitung jumlah tertimbang dari k model (atau pemungutan suara tertimbang).

Algoritma 2 Pengujian Ensemble Deep Learning

Input: Data pengujian $T'\{(t_i'), i = 1,2, \dots, N_3\}$, Kumpulan fitur $Fr = \{F_j, j = 1,2, \dots, k\}$, model-model terlatih M_j dan matriks bobot W_j.

Output: Label prediksi PL_i.

1: **untuk setiap** $t_i' \in T'(i = 1, \dots, N_3)$ **lakukan**

2: **untuk setiap** $F_j \in Fr(j = 1, \dots, k)$ **lakukan**

3: $L_j \leftarrow M_j(t_i', F_j)$;

4: **akhir untuk**

5: $PL_i = \begin{cases} \mathbf{1} & \boldsymbol{jika} \ \sum_{j=1}^{k} L_j * W_j \geq \frac{1}{2}; \\ \mathbf{0} & \boldsymbol{lainnya} \end{cases}$

6: **akhir untuk**

7: **kembalikan** PL_i

Seperti yang terlihat pada Baris 2–4 Algoritma 2, label L_j $(j = 1, \dots, k)$ yang dihasilkan oleh pembelajar lemah j^{th} dihitung untuk setiap instance pengujian. Setelah itu, label akhir PL_i dihitung seperti yang ditunjukkan pada Baris 5 Algoritma 2. Dengan demikian, jika jumlah tertimbang yang dihasilkan lebih besar dari setengah, label diprediksi sebagai positif. Dengan demikian, pembelajar lemah dengan kinerja validasi yang lebih tinggi memiliki dampak yang lebih besar pada prediksi pengujian dalam algoritma ensemble yang diusulkan.

11.4 EVALUASI KINERJA ENSEMBLE DEEP LEARNING

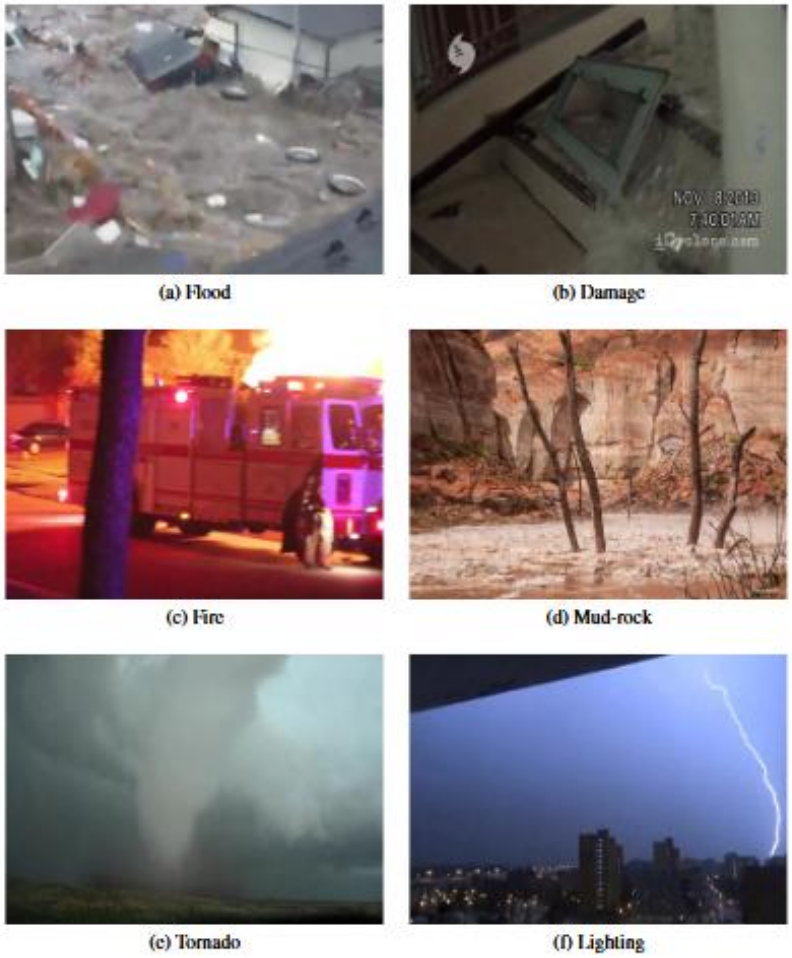
Kerangka kerja *Ensemble Deep Learning* (EDL) yang diusulkan dapat diterapkan secara umum pada berbagai masalah dunia nyata seperti klasifikasi gambar, audio, dan teks. Dalam artikel ini, kami secara khusus mengevaluasi kerangka kerja kami pada dua dataset video untuk mendeteksi peristiwa semantik. Dataset pertama mencakup video yang berisi bencana alam, sedangkan yang kedua adalah dataset video skala besar publik yang disebut TRECVID.

Karena kedua dataset sangat tidak seimbang, metrik biasa seperti akurasi dan kesalahan kuadrat rata-rata mungkin tidak efektif dan dapat diandalkan. Alasannya adalah bahwa pengklasifikasi konvensional, yang sebagian besar bias terhadap kelas mayoritas, mungkin menunjukkan akurasi yang sangat tinggi pada kelas ini sementara kita lebih tertarik pada kelas minoritas. Oleh karena itu, kerangka kerja yang diusulkan dievaluasi menggunakan metrik pengukuran umum untuk data yang tidak seimbang. Secara khusus, parameter matriks kebingungan termasuk *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), dan *False Negative* (FN), serta Precision, Recall, dan ukuran F1 (Persamaan (11.3)) digunakan untuk mengevaluasi kinerja EDL.

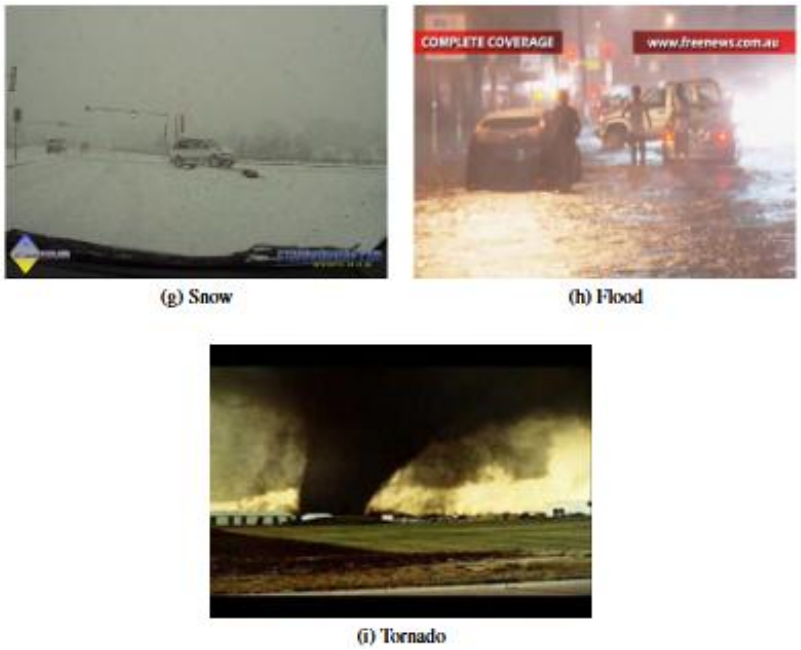
Kerangka kerja pembelajaran mendalam yang digunakan dalam eksperimen berikut disebut Caffe. Ini mencakup teknik pembelajaran mendalam tingkat lanjut dan arsitektur terkini. Keuntungan utama Caffe adalah model referensi pra-terlatihnya yang kaya dan diperbarui, yang dapat dengan mudah digunakan untuk fine-tuning dan transfer learning. Di antara semua model referensi, kami menggunakan model yang paling sukses dalam literatur termasuk R-CNN, CaffeNet, GoogleNet, AlexNet, dan ResNet untuk dimanfaatkan dalam kerangka kerja pembelajaran mendalam ensemble yang diusulkan. Untuk melakukan hal tersebut, kami mengekstrak berbagai set fitur dari keyframe video menggunakan model referensi Caffe. Semua fitur diekstrak dari lapisan fully-connected terakhir (tipe InnerProduct) dari setiap model. Misalnya, lapisan "fc-rcnn" dari R-CNN, lapisan "fc8" dari CaffeNet dan AlexNet, "loss3/classifier" dari GoogleNet, dan "fc1000" dari ResNet digunakan. Semua model awalnya dilatih pada dataset ImageNet, sebuah basis data gambar berskala sangat besar yang mencakup 1000 kelas. Lapisan terakhir dari setiap model yang dipilih menghasilkan vektor fitur berdimensi 1000 kecuali R-CNN, yang menghasilkan 200 fitur pada lapisan terhubung penuhnya.

Evaluasi EDL pada Dataset Bencana

Dataset bencana dikumpulkan dari video YouTube yang mencakup tujuh bencana alam seperti banjir, kerusakan, kebakaran, lumpur-batu, tornado, dan petir.



Gambar 11.2 Contoh kerangka kunci bencana dengan konsep yang dianotasi.



Gambar 11.2 (Lanjutan)

Secara keseluruhan, dataset ini mencakup sekitar 80 video. Setelah menerapkan teknik deteksi batas bidikan video dan pemilihan keyframe, 6884 bidikan diekstrak dari dataset ini. Rasio positif/negatif (P/N) rata-rata dari dataset bencana adalah 0,051. Rasio ini menunjukkan distribusi data yang tidak seimbang. Gambar 11.2 menunjukkan beberapa contoh keyframe dari dataset bencana.

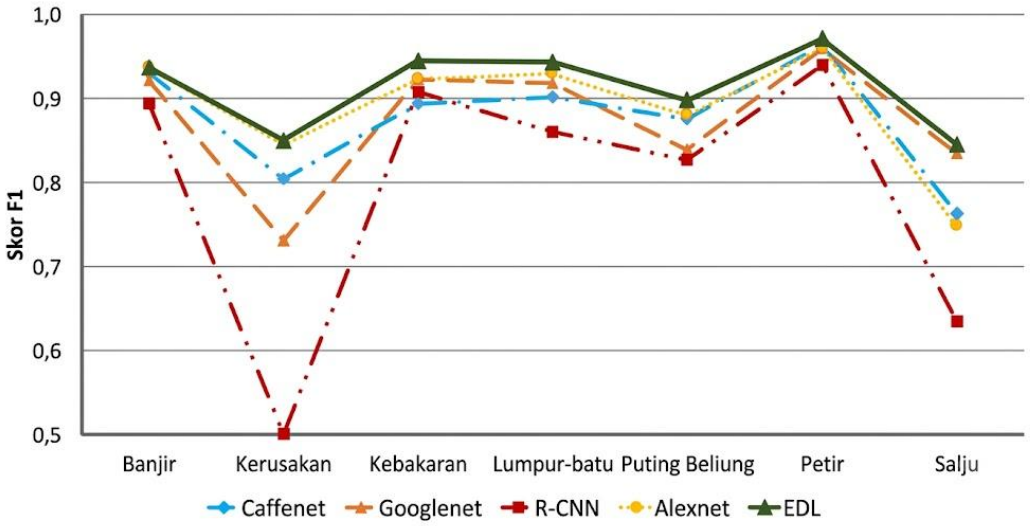
Untuk dataset bencana, kerangka kerja EDL yang diusulkan dibandingkan dengan dua set algoritma: fitur buatan tangan (atau fitur rekayasa) dan fitur pembelajaran mendalam. Untuk kelompok pertama, beberapa fitur tingkat rendah dan menengah seperti HOG, CEDD, histogram warna, tekstur, dan wavelet diekstrak. Kumpulan fitur keseluruhan untuk setiap keyframe mencakup 707 atribut visual. Sedangkan pada kelompok kedua, fitur dihasilkan dengan menerapkan model referensi pembelajaran mendalam secara langsung pada setiap keyframe. Setelah ekstraksi fitur, kami menerapkan beberapa pengklasifikasi terkenal termasuk Pohon Keputusan (DT), Pemeriksaan Korespondensi Berganda (MCA), SVM, dan algoritma boosting pada fitur yang dibuat secara manual. Yang terakhir adalah algoritma ensemble yang dapat dianggap sebagai tolok ukur yang kredibel untuk dibandingkan dengan kerangka kerja ensemble kami. Pengklasifikasi SVM juga digunakan untuk kelompok kedua (fitur mendalam) karena telah terbukti sebagai pengklasifikasi yang sukses ketika diintegrasikan dengan teknik pembelajaran mendalam. Untuk mendapatkan perbandingan yang adil, semua pengklasifikasi disetel untuk mencapai hasil tertinggi pada dataset ini dan dievaluasi melalui validasi silang 3-fold.

Tabel 11.2 menunjukkan presisi, recall, dan skor F1 rata-rata untuk kelompok fitur buatan tangan dan fitur mendalam yang terintegrasi dengan berbagai algoritma klasifikasi. Baris terakhir juga menunjukkan kinerja algoritma EDL yang diusulkan. Seperti yang dapat disimpulkan dari tabel, kerangka kerja yang diusulkan meningkatkan hasil kinerja dibandingkan dengan semua teknik di kedua kelompok. Dengan kata lain, ia tidak hanya mengungguli semua pengklasifikasi konvensional yang terintegrasi dengan fitur rekayasa, tetapi juga mengalahkan jaringan saraf dalam yang terkenal seperti GoogleNet dan AlexNet. Dengan melihat lebih dalam pada hasilnya, dapat disimpulkan bahwa SVM dan teknik ensemble (*boosting*) memperoleh kinerja tertinggi dalam hal skor F1 pada kelompok fitur buatan tangan. Secara khusus, SVM memiliki presisi tertinggi dibandingkan dengan semua algoritma lain termasuk EDL yang kami usulkan. Namun, nilai recall-nya yang rendah menurunkan skor F1 keseluruhannya. Perlu disebutkan bahwa nilai recall yang lebih tinggi, atau dengan kata lain false negative yang lebih rendah, lebih disukai dalam data yang tidak seimbang di mana deteksi yang benar dari kelas minoritas sangat penting (misalnya, dalam aplikasi deteksi kanker). Oleh karena itu, mengintegrasikan fitur-fitur mendalam dengan pengklasifikasi SVM secara wajar dapat meningkatkan nilai recall dan ukuran F1 secara signifikan seperti yang ditunjukkan pada kelompok hasil kedua (fitur pembelajaran mendalam) pada Tabel 11.2. Pada kelompok ini, AlexNet mencapai skor F1 tertinggi dibandingkan dengan teknik pembelajaran mendalam lainnya. Fakta menarik ini menunjukkan bahwa arsitektur yang sangat dalam dan kompleks (misalnya, GoogleNet) tidak selalu berguna untuk berbagai jenis dataset. Terkadang versi jaringan saraf dalam yang lebih ringan tidak hanya lebih efisien daripada yang kompleks, tetapi

juga dapat digeneralisasi untuk berbagai tugas serupa. Misalnya, dalam percobaan ini, meskipun sifat ImageNet sangat berbeda dengan dataset bencana kami, tetapi dapat dilihat bahwa sebagian besar model pra-terlatih (misalnya, AlexNet) pada ImageNet dapat mengklasifikasikan peristiwa bencana dalam video dengan kinerja yang wajar. Akhirnya, kerangka kerja yang diusulkan memanfaatkan kekuatan fitur pembelajaran mendalam yang terintegrasi dengan teknik ensemble baru untuk meningkatkan kinerja keseluruhan dalam hal recall dan skor F1. Skor F1 keseluruhan dihitung sebesar 0.913, yang 4.5% lebih tinggi daripada pengklasifikasi terbaik di grup pertama dan 2.5% lebih tinggi daripada hasil terbaik di grup kedua.

Tabel 11.2 Rata-rata kinerja berbagai set fitur dan pengklasifikasi pada dataset bencana.

Fitur	Pengklasifikasi	Presisi	Recall	Skor F1
handcrafted	DT	0.816	0.823	0.819
handcrafted	MCA	0.894	0.720	0.782
handcrafted	Boosting	0.910	0.841	0.867
handcrafted	SVM	0.957	0.802	0.868
R-CNN	SVM	0.930	0.722	0.794
GoogleNet	SVM	0.918	0.840	0.875
CaffeNet	SVM	0.919	0.840	0.876
AlexNet	SVM	0.924	0.859	0.888
deep features	EDL	0.949	0.883	**0.913**



Gambar 11.3 Evaluasi kinerja untuk berbagai konsep pada kumpulan data bencana.

Gambar 11.3 memvisualisasikan hasil kinerja setiap algoritma pembelajaran mendalam di mana sumbu y mengacu pada skor F1 dan sumbu x menunjukkan berbagai peristiwa bencana. Dapat dilihat dari gambar ini bahwa kerangka kerja EDL yang diusulkan meningkatkan skor F1 untuk semua peristiwa bencana dibandingkan dengan teknik lain. Dalam gambar ini, dalam sebagian besar kasus, R-CNN memiliki kinerja terendah yang dapat disebabkan oleh dua

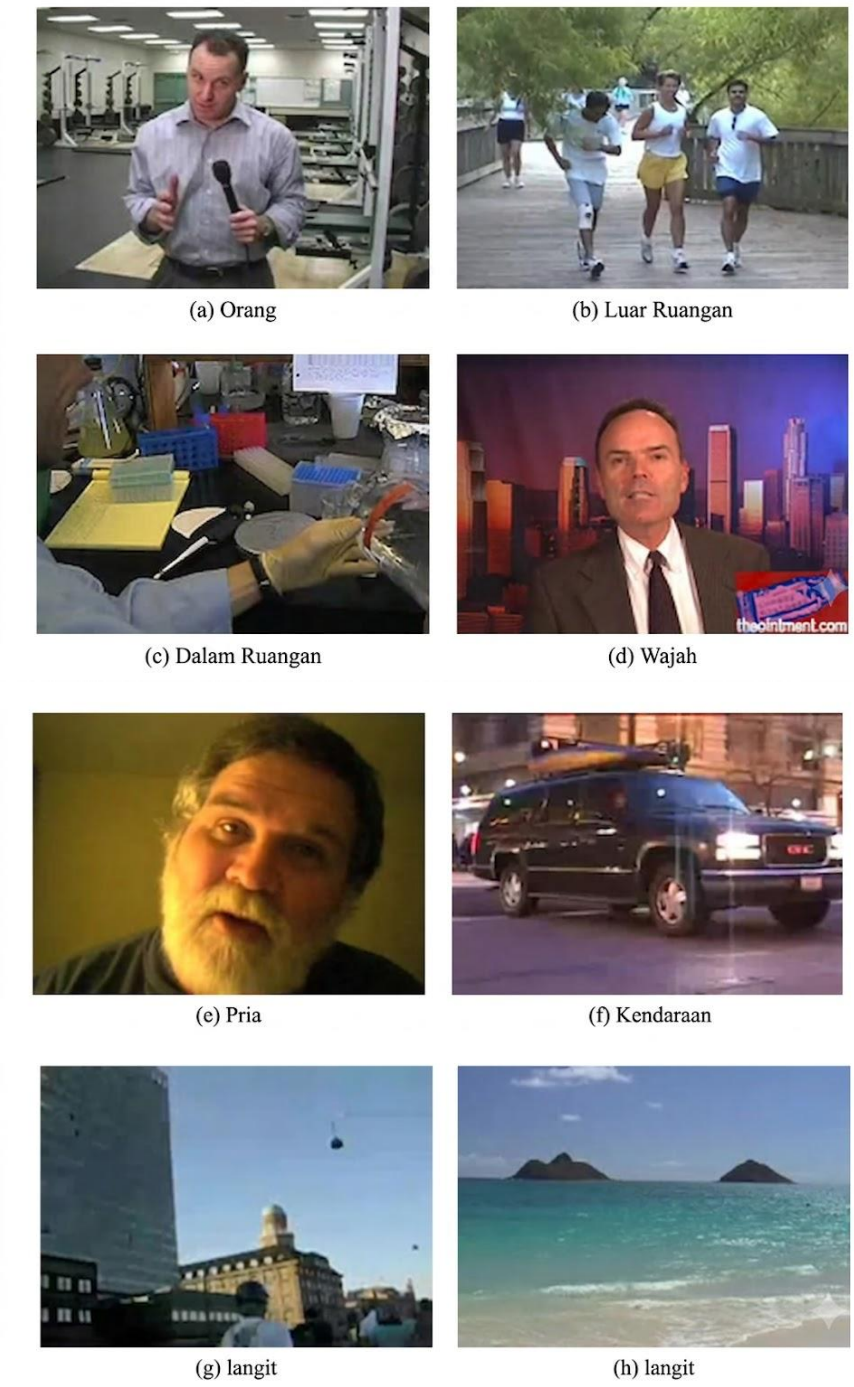
alasan utama. Pertama, arsitekturnya terutama dirancang untuk deteksi objek berbasis wilayah dan segmentasi semantik, sehingga arsitekturnya tidak sesuai dengan tugas deteksi peristiwa video berbasis bingkai. Selain itu, ia menghasilkan 200 fitur yang mencakup informasi lebih sedikit daripada arsitektur pembelajaran mendalam terpilih lainnya yang menghasilkan 1000 fitur di lapisan terakhirnya. CaffeNet dan GoogleNet telah mencapai kinerja rata-rata yang sangat dekat meskipun arsitekturnya berbeda. Lebih spesifiknya, CaffeNet mencapai kinerja yang lebih tinggi untuk konsep tornado dan kerusakan, sementara GoogleNet mengalahkan CaffeNet dalam kebakaran dan salju. AlexNet mengungguli semua teknik pembelajaran mendalam lainnya dalam hal skor F1 untuk hampir semua konsep kecuali petir dan salju. Terakhir, teknik yang kami usulkan berhasil meningkatkan hasil pada semua peristiwa semantik dan mengungguli algoritma pembelajaran mendalam terkini.

Evaluasi EDL pada Dataset TRECVID

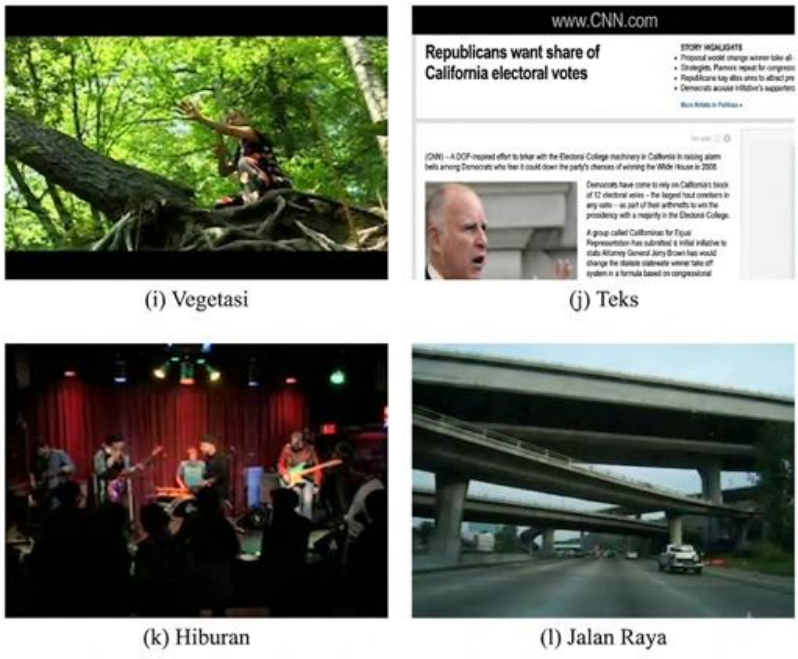
Eksperimen lebih lanjut dilakukan untuk menunjukkan efektivitas kerangka kerja EDL yang diusulkan. Untuk tujuan ini, dataset TRECVID 2011 IACC.1.B yang mencakup video Internet Archive di bawah lisensi Creative Commons dipilih sebagai tolok ukur evaluasi. Dalam tugas pengindeksan semantik (SIN) TRECVID, mirip dengan dataset bencana, tujuannya adalah untuk mendeteksi konsep semantik yang terkandung dalam cuplikan video. Penugasan otomatis label atau tag semantik merupakan langkah mendasar untuk penelusuran, pencarian, dan penyaringan video lebih lanjut, dan lain sebagainya. Secara keseluruhan, dataset IACC.1.B mencakup ratusan ribu keyframe video pelatihan dan pengujian serta 346 konsep. Dalam artikel ini, 20.000 keyframe (10.000 instance pertama dari data pelatihan dan 10.000 instance pertama dari data pengujian) dipilih untuk mengevaluasi kerangka kerja EDL kami. Dalam dataset ini, sebuah konsep mengacu pada konten atau objek semantik tingkat tinggi seperti orang, kendaraan, dan langit. Gambar 11.4 menunjukkan beberapa contoh keyframe dalam dataset ini. Tantangan utama dari dataset ini adalah distribusinya yang sangat tidak seimbang atau miring. Tabel 11.3 menyajikan statistik dari konsep-konsep terpilih dalam set pelatihan dan pengujian. Konsep-konsep ini dipilih karena popularitasnya dan juga variasi rasio P/N yang dimilikinya. Oleh karena itu, kita dapat mengevaluasi perilaku dan fungsionalitas kerangka kerja yang diusulkan dalam berbagai situasi. Misalnya, konsep ``Orang", ``Luar Ruangan", dan ``Jalan" mencakup lebih banyak contoh positif (rasio P/N di atas 15%) dalam set pelatihan dibandingkan dengan konsep lainnya. Rasio P/N rata-rata untuk set pelatihan dan set pengujian masing-masing adalah 0.087 dan 0.039.

Dalam percobaan ini, hasil dari kerangka kerja kami dibandingkan dengan hasil dari *Tokyo Institute of Technology* (TiTech) yang terpilih sebagai kinerja terbaik dalam tugas pengindeksan semantik TRECVID 2011. Kelompok TiTech mengekstrak beberapa fitur tingkat rendah seperti SIFT dan MFCC dari setiap cuplikan video. Oleh karena itu, mereka memanfaatkan fitur buatan tangan termasuk fitur visual dan audio. Selanjutnya, Model Campuran Gaussian (GMM) terstruktur pohon tingkat lanjut diusulkan untuk memodelkan distribusi fitur tingkat rendah menggunakan adaptasi maksimum a posteriori (MAP). Selain itu, mirip dengan dataset bencana, kerangka kerja EDL dibandingkan dengan beberapa fitur mendalam yang terintegrasi dengan pengklasifikasi SVM. Untuk Dalam eksperimen ini, kami

mengganti R-CNN dengan ResNet karena performa R-CNN yang rendah pada eksperimen sebelumnya. Selain itu, CaffeNet dihilangkan karena memiliki arsitektur yang sangat mirip dengan AlexNet. Dengan demikian, kali ini kami hanya memiliki tiga weak learner dalam kerangka kerja EDL kami.



Gambar 11.4 Contoh keyframe TRECVID dengan konsep yang dianotasi.



Gambar 11.4 (Lanjutan)

Tabel 11.4 menunjukkan nilai presisi dan recall dari semua algoritma deep learning serta nilai dari EDL yang diusulkan untuk setiap konsep dalam dataset ini (karena nilai presisi dan recall dari framework TiTech tidak tersedia, maka tidak tercantum dalam tabel ini). Seperti yang disebutkan sebelumnya, metrik recall lebih penting daripada presisi dalam dataset yang tidak seimbang. Dengan demikian, metode yang diusulkan dapat mencapai nilai recall tertinggi (0.436) sambil mempertahankan presisi setinggi mungkin seperti yang ditunjukkan pada Tabel 11.4. Berdasarkan tabel ini, framework EDL yang diusulkan mengungguli semua metode lain dalam hal nilai recall pada sebagian besar konsep. Meskipun ada beberapa konsep seperti "teks" dan "vegetasi" yang memiliki nilai recall lebih tinggi pada metode lain daripada EDL, presisi rendahnya secara signifikan menurunkan skor F1 keseluruhan.

Dengan demikian, metode yang diusulkan kami mencoba untuk menjaga keseimbangan antara kedua metrik ini dan memberikan skor F1 yang lebih tinggi pada konsep-konsep tersebut. Fenomena ini juga dapat dilihat pada Gambar 11.5, yang memvisualisasikan skor F1 untuk semua algoritma benchmark termasuk TiTech dan semua framework deep learning lainnya. Seperti yang dapat disimpulkan dari gambar, EDL yang diusulkan mengungguli semua metode lain di semua konsep kecuali langit. Dalam konsep ini, meskipun EDL yang diusulkan mendeteksi instance positif terbanyak, AlexNet mencapai nilai F1 tertinggi karena false positive-nya yang rendah atau presisinya yang tinggi. Fakta penting lainnya yang dapat disimpulkan dari Gambar 11.5 adalah skor F1 rendah yang dicapai oleh TiTech (hasil terbaik dalam tugas SIN di TRECVID 2011) dibandingkan dengan teknik deep learning. Dengan kata lain, mirip dengan hasil yang diperoleh menggunakan dataset bencana, fitur deep learning (bahkan yang diekstrak dari arsitektur dangkal dan sederhana) mengandung lebih banyak informasi untuk membedakan objek daripada fitur buatan tangan dalam banyak aplikasi.

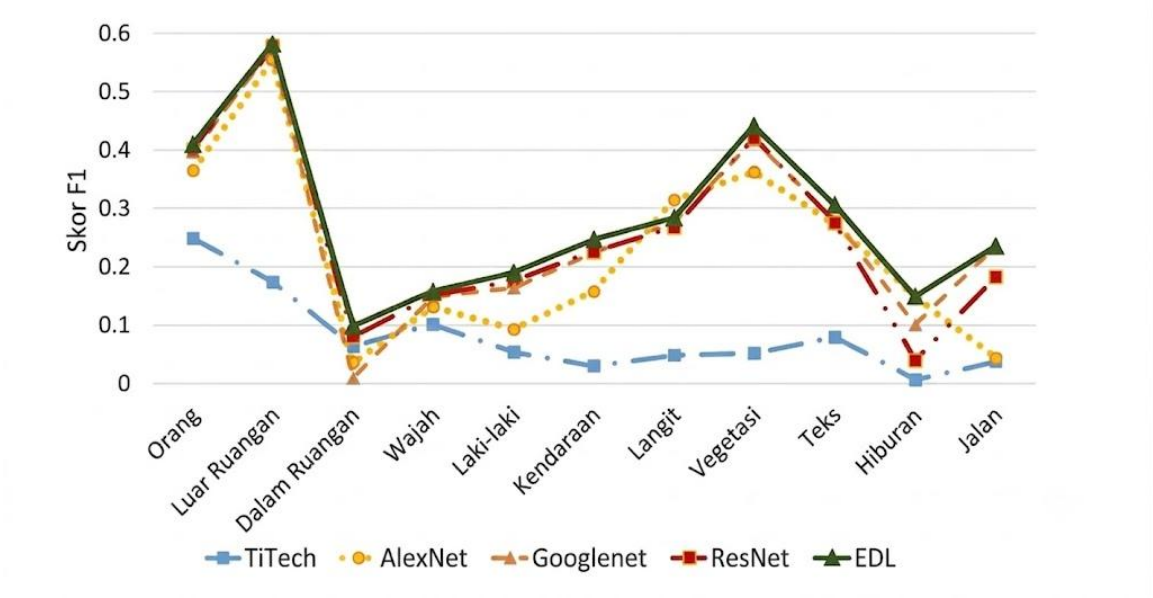
Terakhir, Gambar 11.6 menggambarkan jumlah instance positif yang diprediksi dengan benar (*True Positive*) oleh setiap teknik deep learning. Terlihat bahwa EDL yang diusulkan mendeteksi lebih banyak instance positif dalam banyak konsep seperti "orang", "luar ruangan", dan "dalam ruangan". EDL juga mempertahankan TP setinggi mungkin untuk konsep yang sangat tidak seimbang. Misalnya, konsep "Hiburan" memiliki TP terendah dalam set pengujian (silakan lihat Tabel 11.3), yang berarti hanya sekitar enam instance yang positif di antara 10.000 instance dalam konsep ini. EDL dan AlexNet dapat mendeteksi empat dari enam instance positif, sedangkan GoogleNet dan ResNet hanya dapat mendeteksi satu instance positif dan lima lainnya diklasifikasikan sebagai negatif.

Tabel 11.3 Ringkasan statistik konsep terpilih dalam TRECVID.

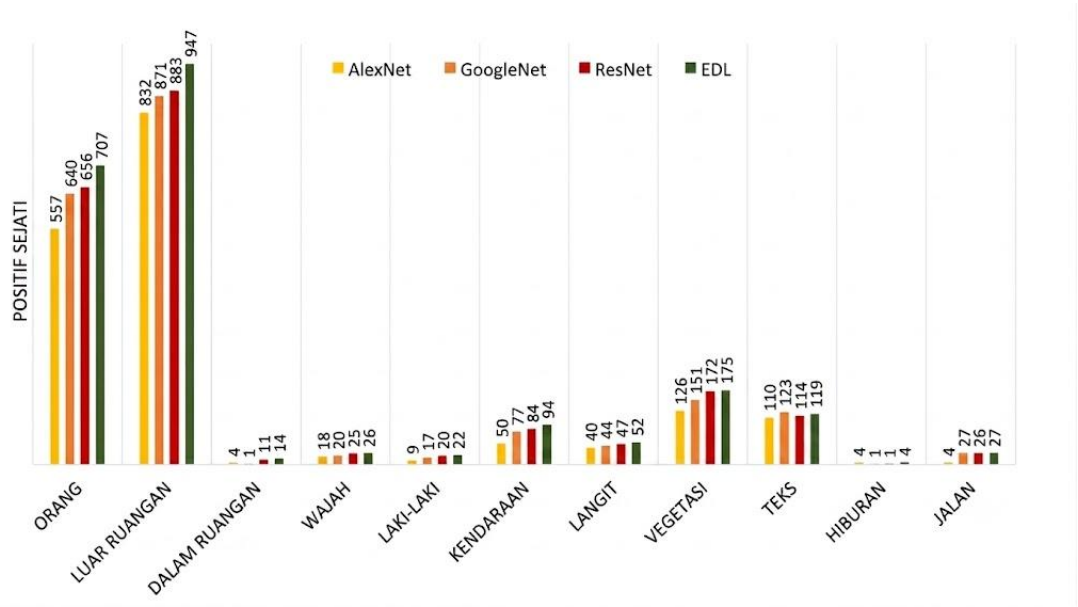
Konsep	Rasio P/N Pelatihan	Rasio P/N Pengujian
Orang	0.3714	0.1511
Luar Ruangan	0.1801	0.1490
Dalam Ruangan	0.0589	0.0218
Wajah	0.0993	0.0035
Laki-laki	0.0418	0.0179
Kendaraan	0.0253	0.0560
Langit	0.0275	0.0119
Vegetasi	0.0675	0.0233
Teks	0.0804	0.0188
Hiburan	0.0413	0.0006
Jalan	0.165	0.0165

Tabel 11.4 Evaluasi kinerja untuk berbagai konsep pada dataset TRECVID.

Konsep	AlexNet		GoogleNet		ResNet		EDL	
	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec
orang	0.321	0.424	0.334	0.487	0.337	0.500	0.332	0.538
luar ruangan	0.488	0.641	0.502	0.672	0.504	0.681	0.483	0.730
dalam ruangan	0.500	0.019	0.333	0.005	0.175	0.052	0.200	0.066
wajah	0.075	0.514	0.085	0.571	0.085	0.714	0.092	0.571
laki-laki	0.529	0.051	0.531	0.097	0.392	0.114	0.393	0.125
kendaraan	0.481	0.094	0.494	0.145	0.393	0.158	0.407	0.177
langit	0.294	0.339	0.220	0.373	0.201	0.398	0.208	0.441
vegetasi	0.269	0.553	0.303	0.662	0.292	0.754	0.328	0.675
teks	0.177	0.595	0.178	0.665	0.178	0.616	0.200	0.643
hiburan	0.083	0.667	0.071	0.167	0.022	0.167	0.083	0.667
jalan	0.167	0.025	0.397	0.167	0.213	0.160	0.397	0.167
Rata-rata	**0.308**	**0.357**	**0.313**	**0.365**	**0.254**	**0.392**	**0.284**	**0.436**



Gambar 11.5 Evaluasi kinerja untuk berbagai konsep pada dataset TRECVID



Gambar 11.6 Perbandingan nilai True Positive untuk berbagai konsep pada dataset TRECVID.

Singkatnya, berdasarkan eksperimen pada dua dataset berbeda dengan distribusi miring, EDL yang diusulkan mencapai kinerja yang menjanjikan dibandingkan dengan teknik terkenal lainnya di bidang ini.

11.5 DETEKSI PERISTIWA VIDEO BERBASIS DEEP LEARNING ENSEMBLE

Karena data multimedia tumbuh secara eksponensial, teknik yang lebih canggih diperlukan untuk menangani sejumlah besar data tidak terstruktur tersebut. Selain itu, deteksi peristiwa video dapat dianggap sebagai salah satu topik paling menantang dalam sistem manajemen multimedia. Dalam artikel ini, kami menargetkan deteksi peristiwa semantik video dengan tujuan menangani dataset besar dan tidak seimbang. Kami juga mempelajari

keuntungan dari penggunaan teknik pembelajaran mendalam untuk pemeriksaan fitur dan pembelajaran transfer. Pertama, beberapa set fitur diekstrak dari algoritma pembelajaran mendalam yang terkenal. Setelah itu, pengklasifikasi mendalam ensemble baru dikembangkan untuk menggabungkan berbagai set fitur mendalam, serta hasil dari setiap pembelajar lemah. Kerangka kerja ini mengurangi masalah data yang tidak seimbang, masalah yang sangat umum dan tak terhindarkan dalam aplikasi dunia nyata. Kerangka kerja yang diusulkan dievaluasi secara ekstensif menggunakan dua dataset video skala besar, yaitu dataset bencana alam dan dataset TRECVID yang populer. Pemeriksaan eksperimental telah dilakukan untuk membandingkan kinerja kerangka kerja EDL yang diusulkan dengan algoritma pembelajaran mesin canggih lainnya. Secara khusus, kinerjanya dibandingkan dengan kelompok fitur buatan tangan dan fitur mendalam, yang terintegrasi dengan beberapa pengklasifikasi terkenal. Berdasarkan hasil eksperimen, kerangka kerja yang diusulkan mengungguli kedua kelompok algoritma tersebut pada dua dataset dengan konsep yang berbeda, yang menunjukkan efektivitas keunggulannya untuk deteksi peristiwa video.

BAB 12
EKSTRAKSI PENGETAHUAN BERBASIS DEEP LEARNING PADA DATA MEDIS

Pembelajaran mendalam memiliki struktur jaringan hierarkis untuk merepresentasikan berbagai fitur data masukan. Metode pembelajaran struktural adaptif *Deep Belief Network* (DBN) dapat mencapai kemampuan klasifikasi yang tinggi sambil mencari struktur jaringan optimal selama pelatihan. Metode ini dapat menemukan jumlah neuron tersembunyi yang optimal untuk data masukan tertentu dalam *Restricted Boltzmann Machine* (RBM) dengan algoritma pembangkitan-penghancuran neuron, dan menghasilkan lapisan tersembunyi baru di DBN dengan perluasan algoritma tersebut. Dalam artikel ini, pembelajaran struktural adaptif DBN (Adaptive DBN) yang diusulkan diterapkan pada data pemeriksaan medis komprehensif untuk prediksi kanker. Sistem prediksi yang dikembangkan menunjukkan akurasi klasifikasi yang lebih tinggi untuk data uji (99,5% untuk kanker paru-paru dan 94,3% untuk kanker lambung) daripada beberapa metode pembelajaran seperti RBM tradisional, DBN, Non-Linear *Support Vector Machine* (SVM), dan *Convolutional Neural Network* (CNN). Selain itu, pengetahuan eksplisit yang membuat proses inferensi DBN terlatih diperlukan dalam pembelajaran mendalam. Pola biner neuron yang diaktifkan untuk input tertentu dalam RBM dan struktur hierarkis DBN dapat merepresentasikan hubungan antara sinyal input dan output. Pola biner ini diklasifikasikan oleh C4.5 untuk ekstraksi pengetahuan. Meskipun pengetahuan yang diekstrak menunjukkan akurasi klasifikasi yang sedikit lebih rendah daripada jaringan DBN yang dilatih, ia mampu meningkatkan kecepatan inferensi sekitar 1/40. Kami melaporkan bahwa aturan IF-THEN yang diekstrak dari DBN yang dilatih untuk data pemeriksaan medis menunjukkan beberapa fitur menarik yang terkait dengan kondisi awal kanker.

12.1 PEMBELAJARAN MENDALAM UNTUK ANALISIS DATA MEDIS

Baru-baru ini, berbagai metode pembelajaran mendalam untuk mempelajari data bervolume besar telah dikembangkan di banyak bidang dan telah dimanfaatkan untuk membuat hidup kita lebih berlimpah. Terutama, teknik pembelajaran mendalam diagnosis citra untuk data medis telah dipelajari secara luas dan cepat. Arsitektur baru *Convolutional Neural Network* (CNN) seperti AlexNet, GoogLeNet, VGG16, ResNet telah dikembangkan untuk pengenalan gambar.

Dalam pembelajaran mendalam, metode coba-coba untuk menemukan struktur jaringan optimal untuk data input yang diberikan akan menjadi tugas yang sulit bahkan bagi perancang yang berpengalaman. Karena biaya komputasi atau sumber daya untuk pemeriksaan big data dari pembelajaran mendalam, pengujian semua kemungkinan set parameter biasanya tidak mungkin dilakukan. Transfer learning adalah metode penggunaan kembali jaringan yang telah dilatih sebelumnya pada masalah baru. Dengan menggunakan jaringan terlatih dengan kemampuan klasifikasi tinggi seperti GoogLeNet, VGG16, atau ResNet,

lebih mudah untuk membangun struktur jaringan untuk data baru. Namun, pembangunan struktur jaringan baru untuk data baru tanpa menggunakan model yang telah dilatih sebelumnya sangat disarankan untuk mencapai kemampuan klasifikasi yang kuat.

Dalam pembelajaran kami, kami telah mengusulkan pembelajaran struktural adaptif dari *Deep Belief Network* (DBN). Metode ini adalah struktur hierarkis dari pembelajaran struktural adaptif *Restricted Boltzmann Machine* (RBM) yang memiliki fungsi pengaturan diri dengan menghasilkan dan menghapus neuron tersembunyi selama pembelajaran. Jumlah RBM juga secara otomatis ditentukan oleh metode pembangkitan lapisan yang memantau kesalahan dan varians dari beberapa parameter. Metode yang dikembangkan menunjukkan kemampuan klasifikasi tertinggi untuk pengenalan gambar dari kumpulan data benchmark seperti MNIST, CIFAR-10, dan CIFAR-100. Rasio klasifikasi untuk kumpulan data hampir 100% untuk kasus pelatihan dan 99,5%, 97,4%, dan 81,2% untuk kasus uji, masing-masing, pada saat penulisan artikel ini.

Dalam artikel ini, sistem prediksi kanker dikembangkan pada data pemeriksaan medis komprehensif yang sangat besar dengan pembelajaran struktural adaptif dari DBN. Data pemeriksaan medis terdiri dari sekitar 100.000 catatan yang tidak hanya berisi tes darah dan pertanyaan tetapi juga gambar medis seperti rontgen dada, CT, dan gambar MRI. Hasil pelatihan metode kami menunjukkan kinerja klasifikasi yang lebih tinggi dibandingkan beberapa metode pembelajaran, yaitu RBM tradisional, DBN, Non-Linear *Support Vector Machine* (SVM), dan CNN.

Ekstraksi pengetahuan yang mengubah pengetahuan implisit dalam pembelajaran mendalam menjadi pengetahuan eksplisit juga merupakan salah satu subjek penting. Untuk ekstraksi pengetahuan dari jaringan terlatih, Jaringan Saraf Buatan Berbasis Pengetahuan (KBANN) adalah metode terkenal yang memetakan "teori domain" spesifik masalah, yang direpresentasikan dalam logika proposisional, ke dalam jaringan saraf dan kemudian menyempurnakan pengetahuan yang diformulasikan ulang ini menggunakan back-propagation. Namun, model ini membutuhkan kerangka pengetahuan untuk membentuk model jaringan dasar. Ichimura mengembangkan KBANN dengan algoritma generasi-penghancuran neuron untuk menemukan jaringan optimal selama pelatihan. Meskipun demikian, model KBANN tidak dapat merepresentasikan banyak aturan untuk big data, karena jaringan saraf multi-lapisan tradisional. Dalam pembelajaran deep learning, metode rekonstruksi jaringan dalam terlatih menjadi jaringan yang lebih kecil dikenal sebagai distilasi pengetahuan. Lee memvisualisasikan fitur tingkat rendah dan tingkat tinggi dari gambar masukan dalam struktur jaringan dalam hierarkis terlatih. Namun, model jaringan besar dalam deep learning tidak dapat menangani ekstraksi fragmen pengetahuan.

Dalam pembelajaran sebelumnya, metode fine-tuning yang memodifikasi bobot jaringan terlatih dikembangkan untuk meningkatkan akurasi klasifikasi untuk pola yang tidak dikenal. Metode ini memanfaatkan input dan neuron tersembunyi RBM yang direpresentasikan sebagai pola biner $\{0, 1\}$. Semua kemungkinan kombinasi nilai biner $\{0, 1\}$ dari input dan output untuk setiap sampel diberikan ke DBN terlatih dan aliran sinyal jaringan dari lapisan bawah ke lapisan atas dihitung. Dengan menyelidiki aliran sinyal, aliran sinyal yang

salah yang terkait dengan output yang salah di-fine-tuning dengan benar. Dalam artikel ini, pengetahuan eksplisit yang direpresentasikan sebagai aturan IF-THEN diekstrak dari jaringan DBN terlatih dengan menggunakan bagian dari metode fine-tuning. Aliran sinyal jaringan yang dikumpulkan untuk semua kemungkinan pola input dan output diklasifikasikan oleh C4.5.

Ekstraksi pengetahuan dari DBN terlatih untuk data pemeriksaan medis telah dilakukan. Pengetahuan tersebut mewakili proses inferensi dari DBN Adaptif yang telah dilatih, yaitu hubungan antara input (tes darah) dan output (probabilitas kanker). Meskipun prediksi berbasis pengetahuan menunjukkan akurasi klasifikasi yang sedikit lebih rendah daripada DBN yang telah dilatih, prediksi tersebut mampu meningkatkan kecepatan inferensi sekitar 1/40. Selain itu, kami melaporkan bahwa pengetahuan yang diekstrak dalam bentuk aturan IF-THEN menunjukkan beberapa fitur menarik yang terkait dengan kondisi awal kanker.

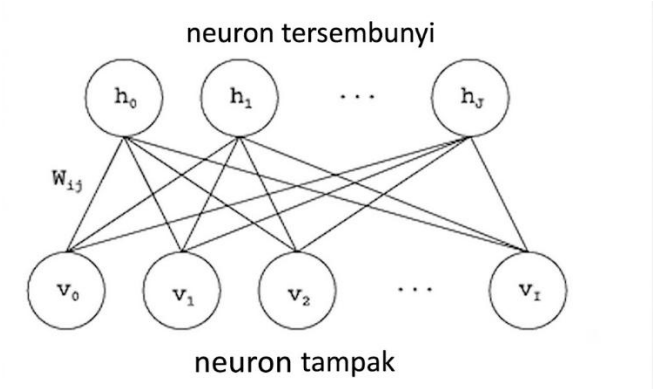
12.2 METODE PEMBELAJARAN ADAPTIF DBN

Bagian ini menjelaskan RBM tradisional dan DBN untuk menggambarkan perilaku dasar metode pembelajaran adaptif DBN yang kami usulkan agar lebih memahami fitur pembelajaran mendalam yang luar biasa.

Restricted Boltzmann Machine

RBM adalah model pembelajaran tak terawasi stokastik. Struktur jaringan RBM terdiri dari dua jenis lapisan biner seperti yang ditunjukkan pada Gambar 12.1: satu adalah lapisan terlihat $v \in \{0,1\}^I$ dengan parameter $b \in \mathbb{R}^{\mathbb{I}}$ untuk pola masukan, dan yang lainnya adalah lapisan tersembunyi $h \in \{0,1\}^J$ dengan parameter $c \in \mathbb{R}^{\mathbb{J}}$ untuk mewakili fitur ruang masukan. I dan J masing-masing adalah jumlah neuron yang terlihat dan tersembunyi. Koneksi antara neuron yang terlihat v_i dan neuron tersembunyi h_j diwakili sebagai bobot W_{ij}. Tidak ada koneksi antar neuron di lapisan yang sama. RBM mencoba meminimalkan fungsi energi $E(v,h)$ dalam pelatihan, yang didefinisikan sebagai berikut, dengan parameter pembelajaran $\theta = \{b, c, W\}$:

$$E(v,h) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j \tag{12.1}$$



Gambar 12.1 Struktur jaringan RBM.

$$p(v,h) = \frac{1}{Z}\exp\big(-E(v,h)\big) \tag{12.2}$$

$$Z = \sum_{v}\sum_{h}\exp\big(-E(v,h)\big) \tag{12.3}$$

di mana b_i dan c_j adalah parameter untuk neuron terlihat v_i dan neuron tersembunyi h_j, masing-masing. W_{ij} adalah parameter (bobot) antara v_i dan h_j. Persamaan (12.2) menunjukkan probabilitas $exp(-E(v,h))$. Z dihitung dengan menjumlahkan energi untuk semua kemungkinan pasangan vektor terlihat dan tersembunyi. Parameter $\theta = \{b, c, W\}$ dioptimalkan untuk data input yang diberikan dengan turunan parsial dari p(v). Cara standar untuk memperkirakan parameter model statistik $p(v)$ adalah estimasi kemungkinan maksimum.

Divergensi Kontrastif (CD) adalah metode pembelajaran RBM yang paling populer sebagai algoritma cepat dari metode pengambilan sampel Gibbs. Pengambilan sampel diimplementasikan untuk keadaan pada rantai Markov dengan Monte Carlo. Karena $exp(-E(v,h))$ adalah nilai diskrit pada v dan h, kami memberikan beberapa diskusi tentang pembelajaran RBM dengan pengambilan sampel biner CD di bawah kontinuitas Lipschitz. Pada pembelajaran sebelumnya, kami menyelidiki perubahan gradien untuk parameter $\theta = \{b, c, W\}$ selama fase pembelajaran, kemudian dua jenis parameter c dan W berpengaruh pada situasi konvergensi RBM kecuali parameter b karena gradien untuk parameter b dapat dipengaruhi oleh berbagai fitur pola input.

Jaringan Kepercayaan Mendalam (Deep Belief Network)

DBN adalah kotak tumpukan untuk pembelajaran tak terawasi stokastik dengan membangun beberapa RBM pra-terlatih secara hierarkis. Struktur jaringan hierarkis DBN dapat dibangun dengan membangun dua atau lebih RBM pra-terlatih. Pola keluaran neuron tersembunyi pada RBM ke-l dapat dilihat sebagai input berikutnya pada RBM ke- $(l+1)$. Probabilitas bersyarat dari neuron tersembunyi j pada RBM ke-l didefinisikan oleh persamaan berikut:

$$p(h_j^l = 1 | h^{\{l-1\}}) = sigmoid\left(c_j^l + \sum_i W_{ij}^l h_i^{l-1}\right) \tag{12.4}$$

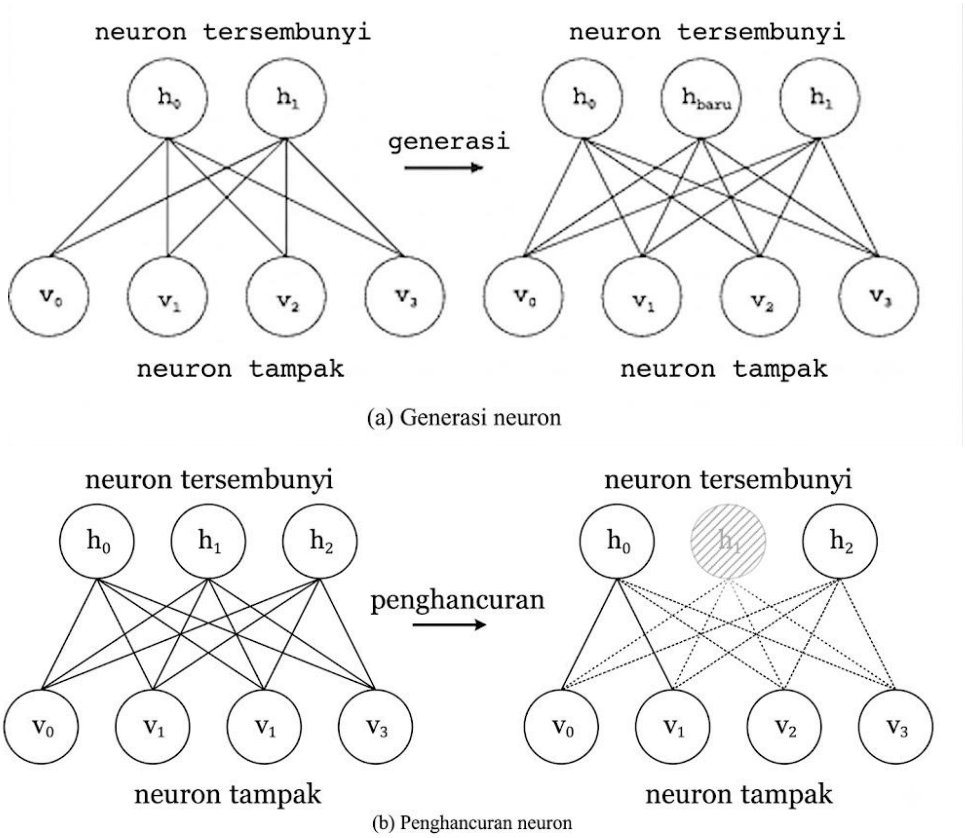
di mana c_j^l dan W_{ij}^l adalah parameter untuk neuron tersembunyi j dan bobot pada RBM ke-l, masing-masing. $h^0 = v$ adalah data input yang diberikan. Ketika DBN yang terlatih melakukan pembelajaran terawasi untuk tugas klasifikasi, lapisan keluaran ditambahkan ke lapisan terakhir, kemudian probabilitas keluaran y_k untuk kategori k dihitung dengan Softmax pada lapisan keluaran sebagai berikut:

$$y_k = \frac{\exp(z_k)}{\sum_j^M \exp(z_j)}, \tag{12.5}$$

di mana z_j adalah pola keluaran dari neuron tersembunyi j pada lapisan keluaran. M adalah jumlah neuron keluaran. Kesalahan antara keluaran y_k dan sinyal guru diminimalkan.

Algoritma Pembangkitan dan Penghancuran Neuron RBM

Secara umum, keputusan struktur jaringan optimal untuk data masukan tertentu bergantung pada keterampilan perancang jaringan, karena banyak percobaan untuk menemukan set parameternya diperlukan untuk mendapatkan kemampuan klasifikasi yang lebih tinggi. Kami telah mengusulkan metode pembelajaran struktural adaptif RBM (Adaptive RBM) dengan mengatur sendiri struktur jaringan untuk data masukan tertentu. Metode ini memperbaiki masalah bahwa RBM tradisional tidak dapat mengubah struktur jaringan neuron tersembunyinya selama pelatihan. Algoritma pembangkitan dan penghancuran neuron, seperti yang ditunjukkan pada Gambar 12.2, dapat menentukan jumlah neuron tersembunyi yang sesuai dengan memantau varians parameter selama pelatihan.



Gambar 12.2 RBM Adaptif.

Algoritma kunci memantau situasi pelatihan jaringan dengan fluktuasi vektor bobot yang disebut Jarak Berjalan (WD). WD dihitung dengan selisih antara varians masa lalu dan varians saat ini untuk parameter pembelajaran seperti bobot selama pelatihan. Jika jaringan saraf tidak memiliki cukup neuron untuk memenuhi syarat inferensi, maka WD cenderung berfluktuasi secara signifikan bahkan setelah periode pelatihan tertentu, karena beberapa

neuron tersembunyi mungkin tidak merepresentasikan pola ambigu karena kurangnya jumlah neuron tersembunyi.

Dalam kasus seperti itu, kita dapat menyelesaikan masalah dengan membagi neuron yang mencoba merepresentasikan pola ambigu menjadi dua neuron dengan mewarisi atribut neuron tersembunyi induk seperti yang ditunjukkan pada Gambar 12.2(a). Kami menggunakan kondisi pembangkitan neuron dengan hasil perkalian varians dari pemantauan dua jenis parameter c dan W kecuali b dalam Adaptive RBM. Alasan pengecualian b adalah karena parameter tersebut berfluktuasi sesuai dengan pola input, karena sinyal input akan mencakup beberapa data noise. Di sisi lain, kita mungkin menemui situasi di mana jaringan memiliki beberapa neuron yang tidak perlu atau berlebihan untuk data input tertentu setelah pembangkitan neuron. Karena neuron-neuron ini tidak digunakan untuk inferensi, jaringan dapat menyebabkan biaya komputasi yang besar. Algoritma penghapusan neuron dapat menghapus neuron yang sesuai berdasarkan aktivasi sinyal output. Neuron yang sesuai akan dihapus seperti yang ditunjukkan pada Gambar 12.2(b) jika kondisi penghapusan neuron terpenuhi selama pelatihan.

Algoritma Pembangkitan Lapisan DBN

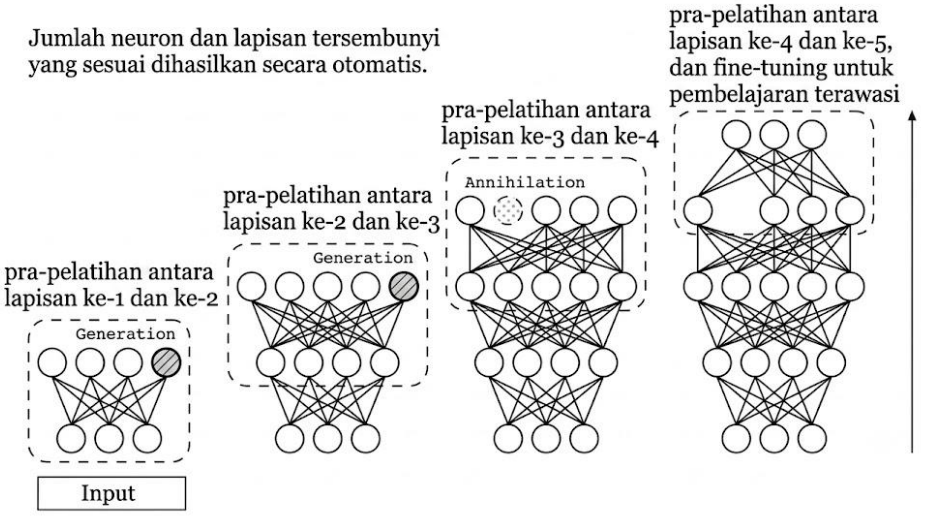
Dengan menggunakan ide RBM Adaptif, kami mengusulkan metode pembelajaran struktural adaptif DBN (Adaptive DBN) yang dapat menentukan jumlah lapisan tersembunyi yang optimal. Secara umum, representasi data DBN melakukan fitur-fitur tertentu dari konsep abstrak ke representasi konkret pada setiap lapisan ke arah lapisan output. Artinya, lapisan bawah memiliki kemampuan representasi non-figuratif, dan lapisan atas membangun objek untuk memahami citra pola input. Adaptive DBN dapat secara otomatis menyesuaikan struktur jaringan optimal melalui pengaturan diri.

Dalam proses pembelajaran Adaptive DBN, kita mengamati WD dan fungsi energi pada setiap lapisan RBM. Jika WD dan fungsi energi secara keseluruhan masih bernilai besar, maka RBM baru diperlukan untuk mengekspresikan struktur jaringan yang sesuai untuk data input yang diberikan. Artinya, WD yang besar dan fungsi energi yang besar berarti bahwa RBM kekurangan kemampuan representasi data untuk memahami citra pola input untuk sinyal pengajaran dalam jaringan secara keseluruhan. Oleh karena itu, kondisi pembangkitan lapisan dengan total WD dan fungsi energi didefinisikan. Gambar 12.3 menunjukkan gambaran umum pembangkitan lapisan dalam Adaptive DBN.

12.3 EKSTRAKSI PENGETAHUAN UNTUK DBN ADAPTIF

KBANN adalah metode yang terkenal untuk mengekstrak pengetahuan berbasis aturan IF-THEN dalam jaringan saraf berlapis banyak. Metode ini pertama-tama merancang struktur jaringan yang mewakili struktur berbasis pengetahuan para ahli. Penerjemahan "aturan ke jaringan" dilakukan dengan memetakan antara kumpulan aturan dan jaringan saraf seperti yang ditunjukkan pada Tabel 12.1. Misalnya, setiap neuron memiliki tingkat pengetahuan yang berbeda. Neuron tersembunyi mewakili asumsi perantara. Bobot mewakili ketergantungan antar neuron. Neuron keluaran mewakili kesimpulan akhir. Struktur jaringan termasuk bobot bergantung pada pemetaan jaringan.

Ichimura menjelaskan KBANN dengan algoritma pembangkitan-penghancuran neuron. Struktur jaringan optimal untuk pengetahuan ditentukan selama pelatihan, dengan memfluktuasikan bobot yang sesuai, pembangkitan neuron, dan penghancuran. Meskipun KBANN merupakan metode yang berguna untuk data kecil berupa nilai numerik, metode ini tidak tepat untuk data besar termasuk gambar, karena sulit untuk menentukan struktur jaringan awal dalam pembelajaran mendalam terkini.



Gambar 12.3 Gambaran Umum DBN Adaptif.

Tabel 12.1 Korespondensi antara basis pengetahuan dan jaringan saraf.

Basis Pengetahuan		**Jaringan Saraf**
Kesimpulan akhir	⇔	Unit keluaran
Fakta pendukung	⇔	Unit masukan
Kesimpulan antara	⇔	Unit tersembunyi
Dependensi	⇔	Koneksi berbobot

Dalam pembelajaran mendalam, metode rekonstruksi jaringan dalam yang telah dilatih menjadi jaringan yang lebih kecil dikenal sebagai distilasi pengetahuan. Distilasi pengetahuan adalah cara sederhana untuk meningkatkan kinerja model pembelajaran mendalam. Pembelajaran mendalam dilatih untuk model jaringan yang kompleks dan besar, dan kemudian model yang telah dilatih dapat mengekstrak fitur-fitur penting dari data yang diberikan. Hinton mengusulkan metode distilasi yang secara signifikan meningkatkan kinerja dalam pembelajaran mendalam dengan mendistilasi pengetahuan dalam kumpulan model menjadi satu model tunggal. Model kecil dapat menghasilkan hasil yang sebanding dan bahkan mampu mereplikasi hasil dari jaringan yang rumit. Namun, model distilasi tidak dapat merepresentasikan pengetahuan eksplisit seperti aturan IF-THEN untuk inferensi.

DBN Adaptif dapat memperoleh struktur jaringan optimal selama pelatihan dengan kemampuan klasifikasi yang lebih tinggi daripada metode pembelajaran mendalam yang ada di beberapa kumpulan data benchmark gambar. Dalam artikel ini, kami mengusulkan metode

ekstraksi pengetahuan untuk inferensi dari DBN Adaptif yang telah dilatih. Pertama, semua kemungkinan kombinasi pola input-output diberikan ke DBN yang telah dilatih untuk menemukan ruang data yang lengkap. Metode ini memanfaatkan input dan neuron tersembunyi RBM yang direpresentasikan sebagai pola biner $\{0, 1\}$. Kedua, aliran sinyal jaringan dari lapisan bawah ke lapisan atas dihitung. Terakhir, pengetahuan eksplisit yang direpresentasikan sebagai aturan IF-THEN diekstrak dari jaringan DBN yang telah dilatih. Aliran sinyal jaringan yang dikumpulkan untuk semua kemungkinan pola input dan output diklasifikasikan oleh C4.5. Dalam artikel ini, pengetahuan tersebut merepresentasikan proses inferensi yang menunjukkan hubungan antara input (tes darah) dan output (probabilitas kanker).

Metode Penyesuaian Halus

Meskipun Adaptive DBN menunjukkan kemampuan klasifikasi yang tinggi, metode ini tidak mampu mengklasifikasikan pola yang tidak dikenal dengan sempurna, karena data masukan dengan pola ambigu menyebabkan klasifikasi yang salah. Dalam kasus seperti itu, metode penyesuaian halus yang memodifikasi sebagian aliran sinyal jaringan berdasarkan pengetahuan akan membantu dalam hal peningkatan kemampuan klasifikasi dan pengurangan biaya komputasi dengan pembelajaran ulang. Aliran sinyal jaringan berarti neuron mana yang diaktifkan di setiap lapisan untuk data masukan yang diberikan.

Ide dasar dari pendekatan ini adalah untuk menghitung frekuensi neuron tersembunyi yang diaktifkan untuk masukan yang diberikan dan untuk menemukan jalur aliran sinyal jaringan dari lapisan masukan ke lapisan keluaran. Setelah aliran sinyal jaringan DBN terlatih dengan semua kemungkinan kombinasi pola biner diperiksa, bobot yang menyebabkan sinyal keluaran yang salah dimodifikasi dengan benar. Algoritma 1 menunjukkan prosedur metode penyesuaian halus yang kami usulkan. Untuk lapisan ke-l, bobot yang menghasilkan pola yang salah dimodifikasi oleh Algoritma 2. Metode ini dapat memperbaiki sebagian jalur sinyal jaringan untuk masukan yang diberikan sesuai dengan pola yang diaktifkan oleh neuron tersembunyi. Kemampuan klasifikasi jaringan terlatih ditingkatkan dengan metode fine-tuning.

Ekstraksi Pengetahuan untuk Inferensi

Dalam artikel ini, pengetahuan eksplisit untuk proses inferensi Adaptive DBN diekstrak dari jaringan terlatih. Idenya didasarkan pada metode fine-tuning. Dengan demikian, aliran sinyal melalui jaringan dari lapisan bawah (dekat dengan input) ke lapisan atas (dekat dengan output) untuk data input yang diberikan diekstrak. Aliran sinyal terdiri dari pola input dan output dalam DBN terlatih. Hubungan antara input dan output dieksplorasi oleh algoritma pohon keputusan.

Prosedur ekstraksi pengetahuan adalah sebagai berikut. Algoritma 3 adalah prosedur ekstraksi aturan.

Algoritma 1. Metode penyempurnaan (*fine-tuning*)

1. Struktur jaringan $L(1 \leq l \leq L)$ lapis dibangun dengan melatih Adaptive DBN.
2. Metode penyempurnaan yang dijelaskan pada Algoritma 2 pada lapisan ke-l diterapkan dari lapisan pertama hingga lapisan ke-L.

Algoritma 2. Prosedur modifikasi bobot untuk lapisan ke-l

1. Misalkan $V = \{v_1, \dots, v_n, \dots, v_N\}$ adalah data masukan, di mana N adalah jumlah pola. Misalkan $Y = \{y_1, \dots, y_n, \dots, y_N\}$ adalah sinyal pengajar (*teacher signal*). Jaringan DBN yang telah dilatih memiliki $L(1 \leq l \leq L)$ lapisan, di mana l adalah lapisan target untuk penyempurnaan.
2. Berikan V ke jaringan DBN yang telah dilatih untuk perhitungan *feed forward*. Dalam perhitungan tersebut, simpan informasi neuron mana yang aktif dari lapisan bawah ke lapisan atas. Misalkan X^T dan X^F masing-masing adalah pola masukan yang benar dan salah untuk Y.
3. Pada lapisan ke-l, jika sebuah neuron j memenuhi Persamaan (12.6), modifikasi bobot yang terhubung ke neuron j menjadi w^{correct}. Persamaan (12.6) adalah rasio dari Act_j^T, yang berarti hanya X^T yang diaktifkan pada neuron j. w^{correct} adalah nilai konstanta (contoh: $w^{\text{correct}} = 1$):

$$\frac{\left|\text{Act}_j^T\right|}{N} \geq \theta^T, \tag{12.6}$$

   di mana $\theta^T(0 \leq \theta^T \leq 1)$ adalah nilai ambang batas.
4. Pada lapisan ke- l, jika sebuah neuron j memenuhi Persamaan (12.7), modifikasi bobot yang terhubung ke neuron j menjadi w^{wrong}. Persamaan (12.7) adalah rasio dari Act_j^F, yang berarti hanya X^F yang diaktifkan pada neuron j. w^{wrong} adalah nilai konstanta (contoh: $w^{\text{wrong}} = 0$):

$$\frac{\left|\text{Act}_j^F\right|}{N} \geq \theta^F, \tag{12.7}$$

   di mana $\theta^F(0 \leq \theta^F \leq 1)$ adalah nilai ambang batas.

Algoritma 3. Ekstraksi aturan inferensi

1. Misalkan $V = \{v_1, \dots, v_n, \dots, v_N\}$ adalah data masukan, di mana N adalah jumlah pola.
2. Berikan V ke jaringan DBN yang telah dilatih untuk perhitungan *feed forward*, kemudian hitung nilai aktivasi dari neuron tersembunyi untuk setiap lapisan.
3. Pada nilai aktivasi di neuron tersembunyi, neuron tersembunyi yang paling sering aktif (*most fired*) untuk setiap lapisan ditentukan sebagaimana ditunjukkan pada Gambar 12.4.
4. Ekstrak aturan dari pola sinyal menggunakan algoritma pohon keputusan (*decision tree*) (contoh: C4.5).

Pertama, untuk input yang diberikan ke Adaptive DBN terlatih, nilai aktivasi neuron tersembunyi dihitung untuk setiap lapisan (dari lapisan input ke lapisan output). Kedua, neuron tersembunyi yang paling sering diaktifkan di setiap lapisan dihitung untuk input yang diberikan. Gambar 12.4 menunjukkan contoh aliran sinyal. Nilai numerik pada Gambar 12.4 berarti indeks neuron tersembunyi. Sebagai contoh, baris pertama pada Gambar 12.4 berarti bahwa neuron diaktifkan melalui jalur ``10"→``77"→``34"→``54"→``54" dari lapisan pertama ke lapisan kelima, dan nilai outputnya adalah ``0". Terakhir, aliran sinyal diklasifikasikan oleh algoritma pohon keputusan apa pun.

```
Header #L1,L2,L3,L4,L5,output
       10,77,34,54,54,0
       75,88,68,82,31,0
Data   6,62,82,32,82,1
       17,62,34,82,11,0
       16,11,82,75,82,1
```

Gambar 12.4 Contoh aliran sinyal (neuron tersembunyi yang diaktifkan) untuk data yang diberikan.

Dalam artikel ini, C4.5 digunakan sebagai algoritma pohon keputusan. C4.5 adalah metode yang terkenal dan populer untuk menghasilkan pohon keputusan. Tentu saja, ada banyak algoritma pohon keputusan seperti ID3, C5.0, atau CART. Di antara algoritma-algoritma ini, himpunan aturan yang dihasilkan oleh C4.5 adalah yang sederhana dengan kinerja klasifikasi yang tinggi. C4.5 memperhitungkan nilai yang tidak tersedia, rentang nilai kontinu, pemangkasan pohon keputusan, derivasi aturan, dan sebagainya. Selain itu, algoritma C4.5 dapat melanjutkan pada sub-daftar yang lebih kecil yang memiliki perolehan informasi (entropi) ternormalisasi tertinggi berikutnya. Oleh karena itu, kami memutuskan untuk menggunakan metode C4.5 untuk membuat pohon keputusan untuk penemuan pengetahuan dari DBN terlatih untuk data pemeriksaan medis yang memiliki beberapa data ambigu.

12.4 EVALUASI ADAPTIVE DBN PADA DATA MEDIS

Pada bagian ini, efektivitas Adaptive DBN yang kami usulkan diverifikasi pada data pemeriksaan medis untuk mengembangkan sistem prediksi kanker. Adaptive DBN menunjukkan kemampuan klasifikasi yang lebih tinggi untuk beberapa kumpulan data benchmark gambar dibandingkan dengan metode yang ada termasuk metode CNN. Dalam artikel ini, kami menantang untuk menerapkan metode yang kami usulkan ke basis data medis sebagai big data kompleks termasuk beberapa data ambigu, dan kemudian pengetahuan yang terkait dengan proses inferensi jaringan terlatih diekstrak.

Data Pemeriksaan Medis

Data medis merupakan kumpulan data pemeriksaan kesehatan yang disediakan oleh ``Asosiasi Lingkungan dan Kesehatan Hiroshima". 100.000 catatan dikumpulkan dari April 2012 hingga Maret 2015. Data diklasifikasikan menjadi data numerik dan kategorikal untuk pemeriksaan kesehatan dan citra medis untuk kanker. Tabel 12.2 dan 12.3 masing-masing menunjukkan item uji data numerik dan kategorikal. ``N/A(%)" adalah rasio nilai yang hilang. “Nilai” pada Tabel 12.3 adalah atribut dan rasio frekuensinya dalam data.

Tabel 12.4 menunjukkan empat jenis citra medis. Dalam artikel ini, dua jenis citra medis, “Paru-paru” dan “Lambung” digunakan untuk prediksi kanker. Sebagai sinyal pelatihan, “normal” atau “abnormal” diberikan kepada setiap citra oleh para ahli medis. 100.000 data pemeriksaan dibagi secara acak menjadi 80.000 kasus pelatihan dan 20.000 kasus uji dengan perbandingan empat banding satu.

Tabel 12.2 Item uji data medis (data numerik).

Nama	Rata-rata	Dev. Std	Min.	Maks.	N/A (%)
Usia	42.455	13.654	10	86	0.0
BMI	22.558	3.146	13.6	31.5	4.8
Tekanan Darah Diastolik (DBP)	119.406	13.713	79	159	1.7
Tekanan Darah Sistolik (SBP)	70.456	10.821	41	100	1.4
Sel Darah Putih (WBC)	5720.186	1442.641	1690	9830	57.2
Sel Darah Merah (RBC)	478.14	43.438	354	602	23.5
Hemoglobin (Hb)	14.66	1.442	10.5	18.8	24.1
Hematokrit (Ht)	43.427	3.87	32.5	54.4	48.7
Trombosit (PLT)	23.061	4.75	9.7	36.4	65.9
GOT	21.792	5.369	6	38	26.5
GPT	19.673	8.543	2	46	28
γ-GTP	28.293	16.424	3	82	29.4
ALP	213.463	55.851	56	371	69.2
LDH	172.97	26.279	99	246	90
Kolinesterase (ChE)	322.151	66.231	135	509	95.5
ZTT	9.066	2.759	1.3	16.8	78.1
Bilirubin Total (TB)	0.821	0.262	0.1	1.6	86.0
Protein Total (TP)	7.164	0.371	6.2	8.1	84.1
Albumin (ALB)	4.422	0.224	3.9	5	90.2
Kolesterol Total (TC)	204.431	33.436	111	298	59.9
LDL	121.327	31.157	33	209	23.4
HDL	63.837	15.983	18	109	23.7
Trigliserida (TG)	97.828	47.61	15	245	27.3
Gula Darah	95.545	10.721	67	126	36.4
HbA1c	5.378	0.263	4.7	6.1	62.2
Asam Urat	5.432	1.376	1.5	9.4	58.3
Kreatinin	0.776	0.144	0.36	1.19	57.6
BUN	13.096	3.005	4.7	21.6	90.9
eGFR	76.366	12.109	42.9	110	76.9
Amilase	83.825	22.58	25	148	84.8
ASO	45.728	35.491	9	167	99.8
CEA	1.723	0.993	0.2	4.7	99.2
Elastase	97.403	23.142	80	167	99.4
CA19-9	6.789	4.633	0.8	20.7	99.7
CA125	10.914	4.87	1.8	24.6	99.9
AFP	3.257	1.25	0.7	6.9	99.9
TTT	1.718	0.937	0.2	4.4	99.2

Tabel 12.3 Butir-butir tes data medis (data kategorikal).

Nama	Nilai	N/A (%)
Jenis Kelamin	"pria" (67,0), "wanita" (33,0)	0,0

Protein Urin	"−" (97,9), "1+" (1,4), "2+" (0,6), "3+" (0,1)	4,4
Darah Samar Urin	"−" (89,3), "±" (0,0), "1+" (6,2), "2+" (2,2), "3+" (2,3)	50,4
Urobilinogen	"−" (98,5), "1+" (0,9), "2+" (0,5), "3+" (0,1)	89,0
Glukosa Urin	"−" (97,9), "1+" (0,8), "2+" (0,7), "3+" (0,6)	4,4
CRP	"−" (97,1), "±" (1,5), "1+" (1,0), "2+" (0,4), "3+" (0,0)	93,2
Antigen HBs	"−" (99,2), "+" (0,8)	93,6
Antibodi HBs	"−" (53,5), "+" (46,5)	97,5
Antibodi HBc	"−" (85,8), "+" (14,2)	99,9
Antibodi HCV	"−" (99,4), "1+" (0,3), "2+" (0,1), "3+" (0,2)	95,0
Pepsinogen	"−" (91,2), "1+" (6,1), "2+" (2,6)	97,0
Pylori	"−" (58,4), "+" (41,6)	98,1

Tabel 12.4 Gambar medis.

Tipe	Bagian Tubuh	Metode	Tes Kanker
Rontgen paru-paru	Paru-paru	Sinar-X	
CT paru-paru	Paru-paru	CT	◯
Rontgen lambung	Lambung	Sinar-X	◯
Mamografi	Payudara	Mamografi (Sinar-X)	◯

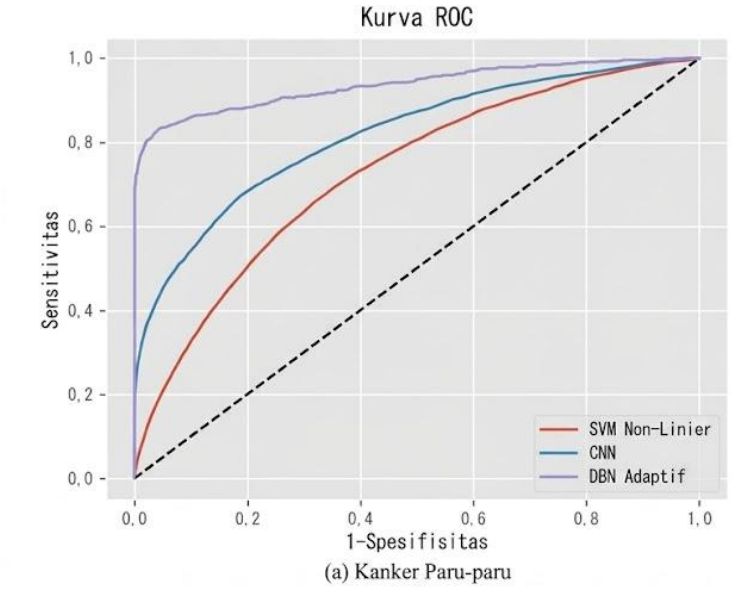
Pengembangan Sistem Prediksi Kanker

Pada bagian ini, efektivitas Adaptive DBN untuk dua jenis kanker diverifikasi. Parameter berikut digunakan untuk pelatihan Adaptive DBN. Algoritma pelatihan adalah metode *Stochastic Gradient Descent* (SGD), ukuran batch adalah 100, laju pembelajaran adalah 0,005, jumlah awal neuron tersembunyi adalah 400, $\theta_G = 0.001, \theta_A = 0.100, \theta_{L1} = 0.05,$ dan $\theta_{L2} = 0.05$. Dua set data input dilatih oleh Adaptive DBN: satu adalah pasangan tes darah dan gambar paru-paru, yang lain adalah pasangan tes darah dan gambar lambung. Untuk membandingkan kinerja Adaptive DBN dengan metode lain, ``Non-Linear SVM", ``CNN", ``Traditional RBM", dan ``Traditional DBN" digunakan dalam percobaan. CNN diimplementasikan pada kerangka kerja ``Keras" dan struktur jaringannya memiliki tujuh lapisan sebagai berikut: konvolusi (Conv2D(64, (3, 3))), konvolusi (Conv2D(64, (3, 3))), pooling (MaxPooling2D(size=(2, 2))), konvolusi (Conv2D(64, (3, 3))), pooling (MaxPooling2D(size=(2, 2))), full-connected (Dense(128)), dan lapisan output dari input ke output. Kami menggunakan komputer dengan spesifikasi berikut selama pelatihan: CPU: Intel(R) 24 Core Xeon E5-2670 v3 2.3 GHz, GPU: Tesla K80 4992 24 GB × 3, Memori: 64 GB, OS: CentOS 6.7 64 bit.

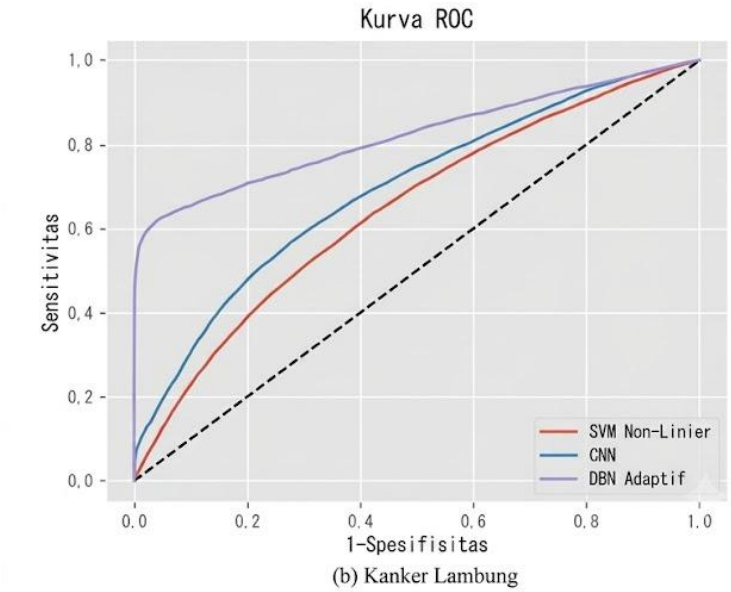
Tabel 12.5 menunjukkan nilai rata-rata dan deviasi standar akurasi klasifikasi pada sepuluh percobaan untuk data uji dalam data medis. Akurasi klasifikasi Adaptive DBN adalah 95,5% untuk kanker paru-paru, 94,3% untuk kanker lambung, yang merupakan nilai lebih tinggi tidak hanya dibandingkan dengan RBM dan DBN tradisional, tetapi juga Non-Linear SVM dan CNN. Gambar 12.5 menunjukkan kurva *Receiver Operating Characteristic* (ROC) untuk dua jenis kanker. Kurva ROC memplot sensitivitas (tingkat positif sejati) pada sumbu vertikal terhadap 1-spesifisitas (tingkat positif palsu) pada sumbu horizontal.

Tabel 12.5 Akurasi klasifikasi untuk data medis.

	Akurasi			
	Paru-paru		Lambung	
Model	**Rata-rata (%)**	**Std.**	**Rata-rata (%)**	**Std.**
Non-Linear SVM	69.6	0.005	65.1	0.005
CNN	93.1	0.005	92.5	0.006
RBM Tradisional	83.3	0.011	82.2	0.010
RBM Adaptif	**85.4**	0.011	**83.8**	0.011
DBN Tradisional	91.1	0.010	91.1	0.010
DBN Adaptif	**95.5**	0.010	**94.3**	0.010



(a) Kanker Paru-paru

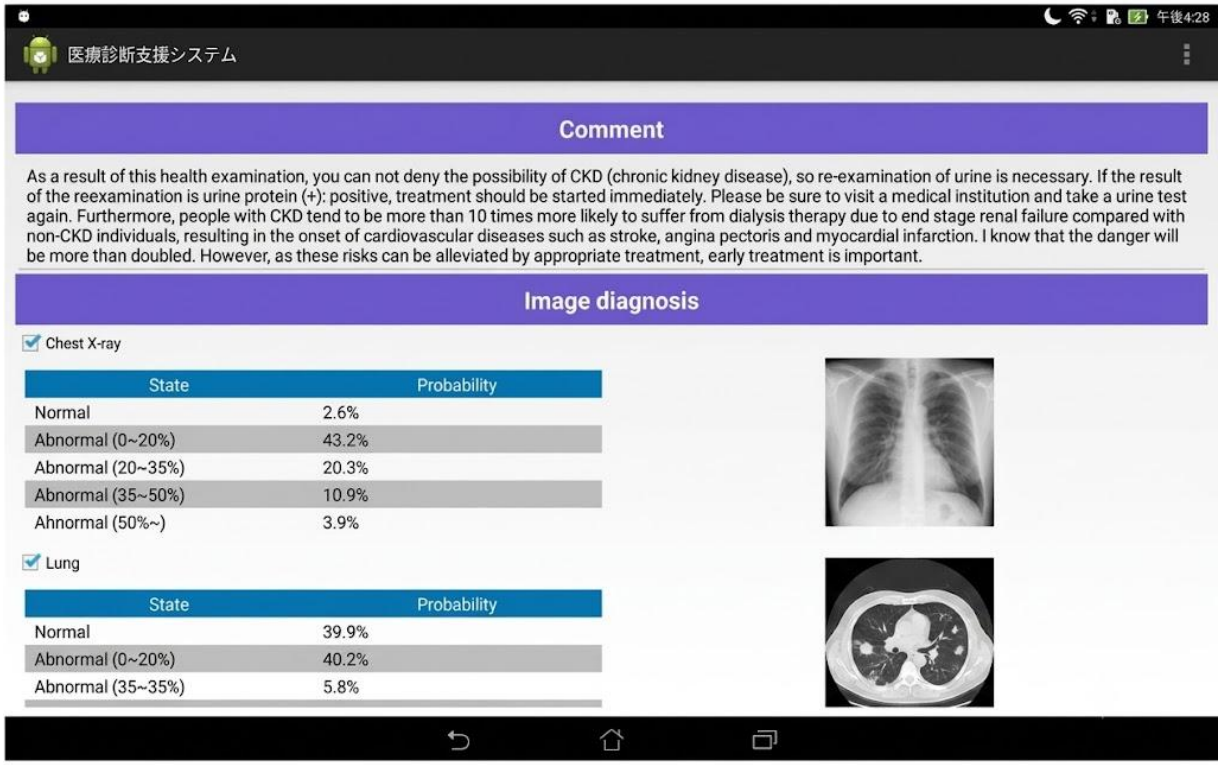


(b) Kanker Lambung

Gambar 12.5 Kurva ROC.

Adaptive DBN menunjukkan kinerja yang baik karena area di bawah kurva Adaptive DBN lebih besar daripada Non-Linear SVM dan CNN. Dengan menggunakan hasil Adaptive

DBN, sistem antarmuka yang dapat menunjukkan probabilitas kanker untuk input tertentu dikembangkan seperti yang ditunjukkan pada Gambar 12.6. Sistem ini dapat memberi perhatian kepada dokter ketika pasien tampaknya memiliki kecenderungan kelainan patologis. Bahkan jika tidak ada ahli medis, misalnya di pulau terpencil, sistem ini dapat menunjukkan perlunya pemeriksaan yang lebih tepat dan pengamatan lanjutan.



Gambar 12.6 Sistem prediksi pemeriksaan kesehatan.

Ekstraksi Pengetahuan dari Jaringan yang Dilatih

Pada bagian ini, hasil eksperimen ekstraksi pengetahuan dari Adaptive DBN yang telah dilatih dilaporkan. Akurasi klasifikasi Adaptive DBN yang telah dilatih adalah 95,5% untuk kanker paru-paru dan 94,3% untuk kanker lambung. Dengan menggunakan metode fine-tuning Algoritma 1 dan 2, akurasi klasifikasi Adaptive DBN ditingkatkan menjadi 98,1% untuk kanker paru-paru, dan 98,0% untuk kanker lambung. Dalam proses fine-tuning, aliran sinyal jaringan dalam grafik diperoleh dalam dua jaringan. Dengan Algoritma 3, aliran sinyal ini termasuk input (tes darah) dan output (probabilitas kanker) diklasifikasikan oleh C4.5.

Gambar 12.7 menunjukkan sebagian dari pohon keputusan yang diekstrak dari C4.5. Jumlah aturan IF-THEN yang diekstrak adalah 110 dan 122 untuk paru-paru dan lambung, masing-masing. Aturan yang diekstrak diterapkan pada proses inferensi alih-alih menggunakan jaringan DBN yang telah dilatih. Tabel 12.6 menunjukkan akurasi klasifikasi menggunakan pengetahuan yang diekstrak dan waktu inferensinya (kolom kedua) untuk data uji. "Adaptive DBN (tanpa pengetahuan)" menunjukkan hasil perhitungan dengan Adaptive DBN yang dilatih pada GPU. "Adaptive DBN (dengan pengetahuan)" adalah hasil dengan pengetahuan yang

diperoleh pada CPU, bukan menggunakan jaringan yang dilatih. Kami menggunakan komputer dengan spesifikasi berikut untuk membandingkan kecepatan perhitungan dengan pengetahuan: CPU = Intel(R) Core(TM) i5-4460 CPU @ 3.20 GHz, GPU: GTX 1080 8 GB, Memori: 8 GB, OS: Fedora 23 64 bit.

```
gamma_gtp <= 78 :
|   egfr <= 92 :
|   |   age <= 48 :
|   |   |   got <= 17 : 5 (9.0/1.0)
|   |   |   got > 17 : 3 (25.0/1.0)
|   |   age > 48 :
|   |   |   plt > 17 : 3 (250.0)
|   |   |   plt <= 17 :
|   |   |   |   tg <= 162 : 3 (57.0/1.0)
|   |   |   |   tg > 162 : 5 (3.0)
|   egfr > 92 :
|   |   plt <= 18 : 5 (4.0)
|   |   plt > 18 :
|   |   |   tp > 7 : 5 (3.0/1.0)
|   |   |   tp <= 7 :
|   |   |   |   urea_nitrogen > 10 : 3 (18.0/1.0)
|   |   |   |   urea_nitrogen <= 10 :
|   |   |   |   |   uric_acid <= 4 : 3 (4.0/1.0)
|   |   |   |   |   uric_acid > 4 : 6 (2.0)
gamma_gtp > 78 :
|   age <= 50 : 6 (10.0)
|   age > 50 :
|   |   plt <= 18 : 5 (3.0)
|   |   plt > 18 : 3 (13.0)
```

Gambar 12.7 Contoh pohon keputusan pada C4.5.

Dari Tabel 12.6, metode yang diusulkan dengan pengetahuan yang diperoleh memiliki kecepatan komputasi yang tinggi, meskipun akurasi klasifikasinya sedikit lebih rendah daripada hasil dengan jaringan yang dilatih. Namun, metode yang diusulkan meningkatkan waktu komputasi sekitar 1/40.

Tabel 12.6 Akurasi klasifikasi pengetahuan untuk data uji.

	Paru-paru		**Lambung**	
Metode	**Akurasi (%)**	**Waktu (dtk)**	**Akurasi (%)**	**Waktu (dtk)**
Adaptive DBN (tanpa *knowledge*)	95,5	0,82	94,3	0,83
Adaptive DBN (dengan *knowledge*)	92,1	0,02	91,0	0,02

Tabel 12.7 Pengetahuan yang diekstrak terkait kanker paru-paru.

ID Aturan	**Kondisi**	**Fitur Kanker**
A-1	Usia kurang dari 50, DAN WBC berada dalam rentang normal.	Kemungkinan kanker lebih rendah dari 20%.
A-2	Usia lebih dari 50, DAN WBC, BMI, Hb, ALP, dan eGFR berada dalam rentang normal.	
A-3	WBC berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 20% dan lebih rendah dari 35%.

A-4	WBC, Albumin, dan Protein Total berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 35% dan lebih rendah dari 50%.
A-5	WBC dan RBC berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 50%.
A-6	Usia lebih dari 80 DAN WBC, BMI, ALP, Albumin, Hb, dan eGFR berada di luar rentang normal.	

Tabel 12.7 dan 12.8 menunjukkan 6 aturan yang diekstrak untuk kanker paru-paru dan 12 aturan untuk kanker lambung, yang aturannya dipilih berdasarkan faktor kepastian (lebih dari 0,95). Dari tabel tersebut, secara keseluruhan terdapat kecenderungan ditemukannya kelainan pada "sel darah putih (WBC)" jika kemungkinan kanker tinggi. Untuk kanker paru-paru, ditemukan kelainan yang berkaitan dengan Albumin dan Protein Total.

Tabel 12.8 Pengetahuan yang diekstrak terkait kanker lambung.

ID Aturan	Kondisi	Fitur Kanker
B-1	Usia lebih rendah dari 50 DAN WBC berada dalam rentang normal.	Kemungkinan kanker lebih rendah dari 20%.
B-2	Usia lebih tinggi dari 50, DAN BMI, Albumin, ALP, eGFR berada dalam rentang normal.	
B-3	WBC berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 20% dan lebih rendah dari 35%.
B-4	WBC, GPT, dan GOT berada di luar rentang normal.	
B-5	WBC dan γGTP berada di luar rentang normal.	
B-6	WBC dan ASO berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 35% dan lebih rendah dari 50%.
B-7	WBC, GPT, GOT, γGTP, dan Kreatinin berada di luar rentang normal.	
B-8	Usia lebih dari 80, DAN WBC, BMI, HbA1c, albumin, dan eGFR berada di luar rentang normal.	Kemungkinan kanker lebih tinggi dari 50%.
B-9	WBC dan Pylori berada di luar rentang normal.	
B-10	WBC dan CA19-9 berada di luar rentang normal.	
B-11	WBC dan Amilase berada di luar rentang normal.	
B-12	WBC dan Elastase berada di luar rentang normal.	

Untuk kanker lambung, ditemukan kelainan yang berkaitan dengan GPT, GOT, dan cGTP. Misalnya, dikatakan bahwa rasio protein dalam darah menurun seiring perkembangan kanker. Evaluasi dengan menggunakan data lain harus diperlukan, tetapi aturan yang diekstrak akan dianggap sebagai pengetahuan medis.

12.5 RINGKASAN DAN EVALUASI METODE ADAPTIVE DBN

Pembelajaran mendalam banyak digunakan dalam berbagai bidang pembelajaran, terutama pengenalan gambar. Dalam pembelajaran kami, Adaptive DBN yang dapat menemukan struktur jaringan optimal untuk data yang diberikan telah dikembangkan. Metode ini menunjukkan akurasi klasifikasi yang lebih tinggi daripada metode pembelajaran mendalam yang ada untuk beberapa kumpulan data benchmark. Dalam artikel ini, Adaptive DBN juga diterapkan pada data pemeriksaan medis. Akurasi klasifikasi Adaptive DBN lebih tinggi daripada RBM, DBN, Non-Linear SVM, dan CNN tradisional, dan kemudian sistem medis untuk memberikan informasi kepada dokter terkait abnormalitas atau normalitas kanker dikembangkan. Selain itu, pengetahuan eksplisit seperti aturan IF-THEN diekstrak dari DBN yang telah dilatih sesuai dengan aliran sinyal jaringan. Meskipun inferensi berbasis pengetahuan menunjukkan akurasi klasifikasi yang sedikit lebih rendah daripada jaringan DBN yang telah dilatih, ia mampu meningkatkan kecepatan inferensi sekitar 1/40. Oleh karena itu, aturan-aturan tersebut memungkinkan perangkat untuk melakukan prediksi berbasis pengetahuan dengan kecepatan prediksi dan akurasi prediksi yang tinggi. Selain itu, aturan-aturan yang diekstrak menunjukkan fitur-fitur menarik untuk menemukan kanker pada tahap awal. Metode yang kami usulkan akan ditingkatkan lebih lanjut untuk kemampuan klasifikasi yang lebih baik dengan mengevaluasi metode ini pada kumpulan data besar lainnya. Meskipun C4.5 digunakan untuk algoritma pohon keputusan dalam percobaan, algoritma lain seperti C5.0, CART, dan sebagainya, akan digunakan untuk evaluasi. Selain itu, verifikasi aturan-aturan yang diekstrak dengan para ahli medis akan diperlukan untuk menunjukkan efektivitas metode kami di masa mendatang.

BAB 13
DETEKSI FRAUD BERBASIS GRAF: TREN, TANTANGAN, DAN PROSPEK

Munculnya pembayaran digital telah menyebabkan perubahan signifikan dalam lanskap kejahatan keuangan. Akibatnya, pendekatan deteksi penipuan tradisional seperti sistem berbasis aturan sebagian besar menjadi tidak efektif. Kecerdasan buatan (AI) dan solusi pembelajaran mesin yang menggunakan prinsip komputasi graf telah mendapatkan minat yang signifikan dalam beberapa tahun terakhir. Teknik berbasis graf memberikan peluang solusi unik untuk deteksi kejahatan keuangan. Namun, implementasi solusi tersebut dalam skala industri pada sistem pemrosesan transaksi keuangan waktu nyata telah memunculkan banyak tantangan aplikasi. Dalam artikel ini, kami membahas kesulitan implementasi yang dihadapi oleh solusi graf generasi saat ini dan selanjutnya. Lebih lanjut, tren kejahatan keuangan dan pembayaran digital menunjukkan tantangan yang muncul dalam keberlanjutan efektivitas teknik deteksi. Kami memeriksa lanskap ancaman dan berpendapat bahwa hal itu memberikan wawasan penting untuk mengembangkan solusi berbasis grafik.

13.1 KOMPUTASI GRAF UNTUK DETEKSI KEJAHATAN KEUANGAN

Pembayaran digital telah mengalami pertumbuhan yang belum pernah terjadi sebelumnya dalam dekade terakhir. Pada tahun 2019 saja, 743 juta transaksi (senilai Rp 1,87 kuantiliun) diproses melalui jaringan pembayaran digital Zelle saja. Ini berarti pertumbuhan tahunan sebesar 57% dalam jumlah total transaksi dan peningkatan volume transaksi sebesar 72%. Antara tahun 2013 dan 2018, pembayaran seluler tumbuh lebih dari 120% (dalam tingkat pertumbuhan tahunan gabungan) di Tiongkok. Demikian pula, di India, volume pembayaran digital meningkat sebesar 61% selama lima tahun terakhir.

Secara global, perbankan seluler dan pembayaran digital telah memberikan kesempatan kepada miliaran orang untuk mengakses layanan keuangan. Selain itu, mereka telah memberikan manfaat praktis bagi konsumen individu, bisnis, dan penyedia layanan keuangan, seperti penghematan waktu, kecepatan, kemudahan penggunaan, biaya transaksi yang lebih rendah, dan kemampuan untuk berkembang. Di sisi lain, skema kriminal telah berkembang pesat untuk mendapatkan keuntungan dari lanskap pembayaran digital yang bergerak cepat. Secara tradisional, deteksi penipuan dan kejahatan keuangan bergantung pada sejumlah besar aturan dan ambang batas statis untuk menandai transaksi yang mencurigakan (misalnya, jumlah lebih besar dari Rp 100 juta). Dalam beberapa waktu terakhir, teknik manual dan berbasis aturan tersebut menjadi tidak efektif karena penipu dengan cepat mengetahui aturan statis dan melewatinya.

Volume kejahatan keuangan global diperkirakan sekitar Rp 14–35 kuantiliun per tahun menurut laporan industri terbaru. Sumber lain memperkirakan pencucian uang sekitar 2–5% dari PDB global (hingga Rp 18,7 kuantiliun), yang sebagian besar tidak terdeteksi. 3,2 juta catatan penipuan diajukan melalui Sistem Komisi Perdagangan Federal AS (FTC) pada tahun 2019 saja; menunjukkan peningkatan sebesar 53% dari tahun 2018. Biaya untuk memerangi

dan memulihkan diri dari penipuan juga meningkat lebih dari 30% sejak tahun 2016. Selain peningkatan kasus penipuan, skema penipuan juga berubah secara signifikan. Dalam beberapa tahun terakhir, dengan adopsi umum kartu kredit dan debit EMV, penipuan kartu secara langsung telah menurun tajam. Namun, lonjakan substansial telah dilaporkan dalam upaya penipuan online. Pada tahun 2020, kasus penipuan kartu kredit online meningkat karena pandemi global dan peningkatan volume transaksi digital yang terkait.

Kejahatan keuangan memiliki efek negatif pada individu dan lembaga keuangan serta efek sistemik seperti konsekuensi negatif pada kesejahteraan negara melalui kinerja makroekonomi. Dalam beberapa tahun terakhir, solusi kecerdasan buatan (AI) dan pembelajaran mesin telah mendapatkan minat yang cukup besar di industri jasa keuangan, termasuk fungsi kepatuhan dan manajemen risiko. Algoritma dan basis data grafik telah lama dianggap sebagai alat penting dalam deteksi penipuan. Banyak pembelajaran telah menunjukkan penggunaan deteksi anomali, aliran jaringan, dan pemeriksaan berbasis sub-grafik yang efektif. Akhir-akhir ini, jaringan saraf graf telah menarik minat. Sebelum aplikasi layanan keuangan, jaringan saraf graf telah diimplementasikan dalam berbagai industri. Potensi yang lebih luas untuk meningkatkan generalisasi dan penalaran relasional telah disoroti oleh sejumlah studi. Dalam deteksi kejahatan keuangan dan penipuan, jaringan saraf graf memberikan peningkatan kinerja dan fleksibilitas.

Namun, implementasi solusi berbasis graf dalam pemrosesan transaksi kehidupan nyata dan sistem deteksi kejahatan menghadirkan tantangan baru. Dalam artikel ini, kami meninjau secara praktis penggunaan komputasi graf dalam aplikasi deteksi kejahatan keuangan. Kami menyoroti kesulitan yang dihadapi organisasi pengembang dalam membangun dan menerapkan solusi berbasis graf dalam sistem pemrosesan transaksi keuangan. Lebih lanjut, kami memeriksa tren kejahatan keuangan yang muncul dan membahas implikasinya pada teknik deteksi. Baik deteksi kejahatan keuangan maupun komputasi graf adalah bidang yang sangat luas dan berkembang pesat. Tujuan artikel ini bukanlah untuk memberikan gambaran umum yang komprehensif tentang bidang-bidang terkait (karena harus dipahami bahwa keduanya akan berubah dengan cepat), tetapi untuk menyoroti poin-poin kesulitan umum dan pertimbangan implementasi praktis yang menyeluruh untuk mengembangkan solusi yang lebih efektif.

13.2 TREN DAN KARAKTERISTIK PENIPUAN KEUANGAN

Bagian ini memberikan pemantauan umum tentang beberapa jenis kejahatan dan penipuan keuangan yang umum. Bagian ini membahas karakteristik utama dan tren yang muncul untuk mengidentifikasi persyaratan untuk teknik deteksi berbasis grafik generasi berikutnya. Tren kejahatan keuangan terkini menggarisbawahi sifat bidang yang terus berubah dan menekankan perlunya solusi inovatif.

Jenis dan Tren Kejahatan Keuangan

Kejahatan keuangan didefinisikan sebagai kejahatan non-kekerasan yang umumnya mengakibatkan kerugian finansial. Kejahatan keuangan dapat dikategorikan menjadi empat jenis utama: korupsi, penipuan, pencurian, dan manipulasi. Artikel ini berfokus pada tiga

kategori terakhir dan teknik deteksi yang sesuai. Skema kejahatan keuangan telah mengalami perubahan besar akhir-akhir ini. Menurut survei industri terbaru, hampir semua jenis penipuan telah mengalami peningkatan yang serius. Penipuan eksternal telah meningkat, baik dalam hal volume maupun jumlah total transaksi masing-masing sebesar 61% dan 59%.

1. Penipuan Pembayaran

Penipuan pembayaran mencakup aktivitas kriminal di berbagai saluran pembayaran termasuk transaksi kartu kredit dan debit, ATM, transaksi antar individu (P2P), transfer kawat, transaksi kliring otomatis, pembayaran online, pembayaran tagihan otomatis, cek, dan setoran. Dalam beberapa tahun terakhir, peningkatan yang meluas telah terlihat dalam penipuan di semua saluran pembayaran.

Menurut J. P. Morgan dan Asosiasi Profesional Keuangan, penipuan pembayaran saat ini dilaporkan terjadi di lebih dari 80% organisasi dan terus menunjukkan tren peningkatan. Penipuan kartu (termasuk penipuan kartu kredit dan debit) adalah salah satu segmen terbesar dalam penipuan pembayaran, dengan kerugian yang mencapai Rp 18 triliun pada tahun 2016 menurut Bank Sentral Eropa. Penggunaan kartu chip dan PIN telah menyebabkan pergeseran taktik penipuan dari transaksi kartu hadir (CP) ke transaksi kartu tidak hadir (CNP). Meskipun telah diperkenalkan 3D-Secure, penipuan CNP terus meningkat dan mencapai lebih dari 70% dari total kerugian penipuan kartu. Semua saluran pembayaran online (dari pembayaran tagihan hingga P2P dan pemrosesan cek online) mengalami pertumbuhan yang signifikan dalam beberapa tahun terakhir.

Selain perpindahan cepat ke saluran online dan perubahan taktik penipuan, sering terjadi persilangan antar saluran pembayaran dan jenis penipuan. Sebagai contoh, laporan Bank Dunia menunjukkan bahwa para penjahat semakin banyak menggunakan saluran pembayaran seluler untuk pencucian uang. Karena skala ekonomi menarik bagi organisasi kejahatan keuangan, mereka terus mengejar peretasan pedagang besar untuk mendapatkan akses ke informasi terkait kartu untuk puluhan atau ratusan juta akun, yang memicu penipuan online.

2. Pencurian Identitas

Pencurian identitas mengacu pada proses pencurian informasi pribadi seseorang untuk melakukan penipuan. Skema pencurian identitas menggunakan daftar taktik yang selalu berubah, mulai dari perangkat skimming ATM hingga phishing, smishing, dumpster diving, dan jaringan nirkabel yang dikompromikan. Setelah pencurian identitas itu sendiri, pelaku biasanya menggunakan informasi yang dikompromikan di berbagai saluran. Pencurian identitas dan penipuan akun baru menyebabkan kerugian finansial yang lebih besar dibandingkan dengan jenis penipuan pembayaran lainnya karena waktu yang dibutuhkan untuk mendeteksinya. Hampir 58% kasus pencurian identitas ditemukan setelah empat bulan. Lebih lanjut, sekitar 25% kasus ditemukan setelah dua tahun. Periode penemuan yang panjang seperti itu menguntungkan organisasi kejahatan keuangan dan memotivasi investasi lebih lanjut dalam skema pencurian identitas.

Sejak tahun 2000, telah terjadi pertumbuhan yang stabil dalam jumlah laporan dan pengaduan pencurian identitas. Akhir-akhir ini, pencurian identitas telah menjadi salah satu

jenis penipuan teratas dalam pengajuan kriminal Komisi Perdagangan Federal. Pencurian identitas biasanya dikaitkan dengan kumpulan jenis kejahatan lainnya. Penipuan kartu kredit adalah jenis penipuan teratas untuk penipuan hilir pada tahun 2019, dengan lebih dari 200 ribu kasus yang diajukan. Penipuan rekening kredit baru tumbuh sebesar 88% selama periode yang sama. Demikian pula, penipuan cek dan penipuan keuangan sering dikaitkan dengan pencurian identitas.

3. Penipuan Keuangan

Penipuan keuangan mengacu pada berbagai macam penipuan dan aktivitas kriminal yang menggunakan tipu daya sebagai cara untuk mencuri dari individu yang menjadi target. Kejahatan ini menggunakan taktik yang terus berkembang seperti penipuan telepon, penipuan dukungan teknis, penipuan lansia, penipuan amal dan lotere, penipuan tiket, dll. Setiap kategori kemudian mencakup daftar panjang taktik, seperti penipuan lansia yang disesuaikan yang melibatkan asuransi, penipuan peniruan identitas IRS, hipotek, dan penipuan kakek-nenek, dll. Karena peningkatan populasi lansia, penipuan keuangan yang menargetkan lansia menjadi lebih menonjol dalam beberapa tahun terakhir. Biro Investigasi Federal melaporkan bahwa 80% dari semua penipuan di AS menargetkan populasi berisiko antara usia 55 dan 85 tahun. Penipuan keuangan biasanya terkait dengan pencurian identitas dan pengambilalihan akun, serta jenis penipuan hilirnya di berbagai saluran. Penipuan keuangan telah menjadi salah satu perhatian utama dalam lanskap penipuan. Memeriksa interkonektivitas dan pola jaringan bersama merupakan kunci dalam mendeteksi penipuan keuangan dan kasus penipuan hilir.

4. Pengambilalihan Akun

Penipuan pengambilalihan akun terjadi ketika pelaku mendapatkan akses ke rekening bank korban secara ilegal. Selama ATO, penjahat biasanya juga mengubah kredensial login akun dan informasi kontak, sehingga korban tidak dapat mengakses akun tersebut. Mereka akhirnya menguras dana melalui satu atau lebih saluran pembayaran (seperti transfer kawat). ATO memiliki hubungan yang kuat dengan keamanan siber karena pelaku secara teratur menggunakan pelanggaran data massal, pembajakan SIM seluler, perangkat dan jaringan yang dikompromikan untuk memicu serangan mereka. Teknologi otentikasi yang ditingkatkan, keamanan titik akhir, dan praktik keamanan siber lainnya memainkan peran kunci. Mirip dengan pencurian identitas, ATO menyediakan gerbang ke berbagai jenis penipuan hilir.

ATO telah mengalami peningkatan 78% pada tahun 2019. Selain itu, laporan berita terbaru menunjukkan bahwa insiden ATO meroket antara kuartal pertama dan ketiga selama pandemi global pada tahun 2020. Tingkat yang mengkhawatirkan serupa dilaporkan antara tahun 2019 dan 2020 dari tahun ke tahun. Baru-baru ini, topologi kejahatan serta saluran-saluran yang digunakan untuk melakukan kejahatan telah berubah dengan cepat. Kejahatan keuangan, penipuan, dan ancaman keamanan siber dengan cepat saling berkonvergensi. ATO memberikan contoh utama dari konvergensi ini dan tantangan dalam mendeteksi aktivitas kriminal. Taktik kejahatan siber seperti spear-phishing, back-door, perangkat yang terinfeksi, peretasan ATM, dan perubahan saldo semakin menonjol dalam serangan layanan keuangan. Hal ini semakin mengancam mekanisme pertahanan layanan keuangan tradisional yang secara bertahap menjadi tidak efektif dalam mengatasi perubahan dan konvergensi tersebut. Di sisi

lain, karena organisasi kejahatan keuangan mencari efisiensi dalam serangan berulang, hal ini memberikan peluang signifikan dalam deteksi penipuan melalui teknik berbasis grafik dan integrasi pertahanan siber dengan deteksi kejahatan keuangan.

5. Penipuan Identitas dan Akun Sintetis

Penipuan akun sintetis didasarkan pada identitas palsu yang dibuat agar terlihat seperti pelanggan sungguhan dengan skor kredit dan karakteristik yang menguntungkan. Kasus penipuan identitas sintetis secara teratur menggunakan nomor jaminan sosial dan nomor privasi kredit yang dicampur dengan informasi nyata dan sintetis dari satu atau lebih individu. Dalam beberapa tahun terakhir, telah terjadi pertumbuhan yang signifikan dalam penipuan identitas dan akun sintetis. Laporan terbaru menyoroti bahwa jenis penipuan ini telah tumbuh sekitar 35% dari tahun ke tahun. Salah satu alasan tren ini adalah fakta bahwa penipuan identitas/akun sintetis bersifat non-transaksional. Oleh karena itu, dibutuhkan waktu lebih lama untuk mendeteksi dan biasanya tidak memiliki jalur pelaporan kejahatan tradisional yang diandalkan oleh jenis penipuan lainnya (misalnya, melaporkan transaksi kartu kredit yang curang). Oleh karena itu, keuntungan finansial yang dihasilkan bagi organisasi kriminal secara proporsional lebih tinggi karena waktu deteksi yang lebih lama.

6. Pencucian Uang

Pencucian uang didefinisikan sebagai transfer uang atau dana yang diperoleh secara ilegal yang dihasilkan dari aktivitas kriminal (seperti perdagangan narkoba, pendanaan terorisme, dll.) untuk menyembunyikan sumber sebenarnya. Pencucian uang dapat terjadi dengan atau tanpa sepengetahuan perusahaan keuangan atau pihak lawan yang terlibat dalam transaksi keuangan. Pencucian uang biasanya terdiri dari tiga tahap utama yang masing-masing memerlukan teknik khusus untuk deteksi: (i) Penempatan: Hasil dari kegiatan ilegal ditempatkan ke dalam sistem keuangan. Dana tersebut sering kali distrukturkan menjadi jumlah yang lebih kecil, ditempatkan ke dalam beberapa rekening, di sejumlah lembaga keuangan untuk mencegah deteksi. Para penjahat biasanya melakukan setoran tunai kecil, membeli instrumen moneter/wesel bank agar tetap tidak terdeteksi oleh peraturan dan undang-undang anti pencucian uang. (ii) Pelapisan: Selama tahap pelapisan, terjadi beberapa transfer antara perusahaan cangkang dan individu untuk menyembunyikan asal usul dana ilegal. Anti pencucian uang (AML), kenali-klien Anda (KYC), dan penanggulangan pendanaan terorisme (CTF) telah menjadi fungsi yang sangat penting untuk mencegah transfer tersebut di lembaga keuangan. Namun, upaya anti pencucian uang telah menghadapi kesulitan serius selama beberapa tahun terakhir. (iii) Integrasi: Selama tahap integrasi, dana diintegrasikan ke dalam sistem keuangan seolah-olah dana tersebut legal. Tahap ini dapat melibatkan investasi (seperti real estat atau sekuritas), pembelian barang mewah, dll.

Secara global, diperkirakan Rp 7,15 hingga 18,7 kuantiliun dicuci menurut laporan industri. Selama bertahun-tahun, jumlah dan variasi transaksi yang digunakan dalam pencucian uang menjadi semakin kompleks. Transaksi tersebut seringkali melibatkan sejumlah besar rekening, jenis transaksi, banyak lembaga keuangan dari berbagai yurisdiksi, dan meningkatnya penggunaan lembaga keuangan non-bank (seperti layanan pencairan cek, asuransi, dll.). Kompleksitas ini membuat upaya deteksi menjadi jauh lebih menantang.

7. Jenis Penipuan Lainnya

Penting untuk dicatat bahwa penipuan dan kejahatan keuangan mencakup spektrum luas aktivitas kriminal. Dalam artikel ini, kami menyoroti beberapa jenis berdasarkan prominensi dan tren yang muncul. Penipuan hipotek dan penipuan pinjaman cukup menonjol dalam lanskap penipuan secara keseluruhan. Penipuan ini umumnya melibatkan kompromi rekening pelanggan atau informasi identitas pribadi (PII), seringkali terkait dengan pencurian identitas. Oleh karena itu, baik pelanggan maupun lembaga keuangan menderita kerugian. Terlepas dari pentingnya dalam lanskap penipuan secara keseluruhan, penipuan hipotek telah mengalami penurunan selama beberapa tahun terakhir. Kecurangan internal melibatkan karyawan jasa keuangan yang terlibat dalam aktivitas kriminal termasuk namun tidak terbatas pada pelanggaran sanksi keuangan, perdagangan orang dalam, pengaturan tender, penetapan harga, alokasi pasar, dan penyuapan. Seperti yang telah disorot sebelumnya dengan contoh-contohnya, kecurangan eksternal dapat dikategorikan sebagai kecurangan pihak pertama (melibatkan klien bank sendiri) atau kecurangan pihak ketiga (melibatkan kontak eksternal dan organisasi kriminal). Solusi kecurangan internal juga bergantung pada pemeriksaan interkonektivitas berbasis entitas yang digunakan untuk mendeteksi kecurangan eksternal pihak pertama dan ketiga. Kerugian finansial dan kepatuhan terhadap peraturan berperan sebagai faktor pendorong penting dalam mendeteksi kecurangan internal.

Karakteristik Bersama

Seperti yang telah disebutkan sebelumnya, taktik kejahatan keuangan telah menunjukkan tingkat adaptabilitas yang tinggi terhadap lanskap digital yang berkembang, konvergensi yang signifikan dengan ancaman keamanan siber, dan peningkatan kecenderungan terhadap mekanisme efisiensi. Sebagai contoh, meningkatnya pencurian identitas dan ATO serta penurunan penipuan kartu kredit tatap muka (CP) menyoroti dorongan efisiensi ini. Taktik kejahatan dengan cepat beradaptasi dengan kerentanan sistem pemrosesan pembayaran serta langkah-langkah pencegahan.

Taktik penipuan menunjukkan tingkat kustomisasi yang luar biasa terhadap saluran individual dan mode multi-saluran. Saluran-saluran tersebut menampilkan perbedaan yang mencolok dalam karakteristik penipuan, seperti jenis transaksi, jumlah, waktu pemrosesan, perangkat, persyaratan otentikasi, dan lain-lain. Misalnya, penipuan ATM sangat berbeda dengan penipuan pembayaran tagihan online (dalam hal frekuensi, jumlah, rentang waktu transaksi dan pemrosesan, pihak yang terlibat, kompromi akses, perangkat yang terlibat, dan lain-lain). Karakteristik unik ini memainkan peran penting dalam efektivitas solusi algoritmik. Dalam banyak kasus, teknik yang berkinerja baik di satu saluran menghadapi tantangan kinerja di saluran lain karena perbedaan tersebut (misalnya, seberapa dinamis lingkungannya, skala, atribut, dan lain-lain). Lebih lanjut, kemampuan untuk menyeimbangkan pola spesifik saluran dengan kebutuhan untuk mengintegrasikan teknik untuk penipuan multi-saluran tetap menjadi tantangan.

Menurut Interpol, sebagian besar kejahatan keuangan bersifat lintas batas dan transnasional, melalui penggunaan platform digital dan internet. Selain peningkatan kemampuan untuk menyembunyikan identitas mereka di berbagai wilayah geografis,

organisasi kejahatan keuangan mengandalkan pola transaksi yang kompleks dan operasi geografis yang beragam. Penipuan transnasional juga memberikan kemampuan kepada organisasi kriminal untuk mentransfer dana ilegal ke lokasi dengan pengawasan yang lebih longgar. Meskipun penipuan online tetap menjadi tantangan yang signifikan, komputasi graf memberikan keuntungan yang signifikan dalam mencapai pemeriksaan berbasis interkonektivitas tersebut dalam skala global.

Peran Lembaga Keuangan

Perusahaan keuangan memainkan peran kunci dalam deteksi dan pencegahan kejahatan keuangan. Departemen Keuangan memperkirakan bahwa 99,9% uang hasil kejahatan yang diserahkan untuk deposito di Amerika Serikat masuk ke rekening yang aman. Suatu lembaga keuangan dapat terlibat dalam kejahatan keuangan sebagai korban, pelaku, atau sebagai instrumen. (i) Korban: Perusahaan keuangan sering menghadapi kerugian akibat penipuan (seperti penyalahgunaan informasi dalam aplikasi kredit dan pinjaman identitas sintetis). Penipuan pihak ketiga merupakan salah satu bagian terbesar dari kejahatan keuangan dalam hal volume dan nilai moneter. (ii) Pelaku: Seperti yang disebutkan dalam Pasal 2.1, perusahaan keuangan dapat terlibat langsung dalam kejahatan keuangan melalui penipuan internal, transaksi sendiri, kolusi, skema penggelapan, dll. (iii) Instrumen: Lembaga keuangan juga dapat berfungsi sebagai perantara dan instrumen dalam mentransfer dana dan melakukan transaksi, baik secara sadar maupun tidak sadar. Pencucian uang sering melibatkan bank sebagai instrumen. Meskipun tanggung jawab hukum dapat bervariasi tergantung pada jenis kejahatan keuangan, perusahaan keuangan telah menghadapi kerugian akibat penipuan yang signifikan dan telah melakukan investasi besar dalam beberapa waktu terakhir untuk meningkatkan sistem deteksi dan pencegahan mereka.

Hukum dan Peraturan Kejahatan Keuangan

Selama bertahun-tahun, banyak hukum dan peraturan telah diberlakukan untuk kejahatan keuangan. Undang-Undang Kerahasiaan Bank (1970) menetapkan serangkaian persyaratan pencatatan dan pelaporan pertama oleh individu, bank, dan perusahaan keuangan lainnya untuk mencegah pencucian uang. Tujuan utamanya adalah untuk menilai sumber, volume, dan arus mata uang dan instrumen moneter yang masuk/keluar dari AS atau di lembaga keuangan. Hal ini mengharuskan lembaga keuangan untuk (i) Melaporkan setiap transaksi tunai di atas Rp 100 juta; (ii) Mengidentifikasi pihak-pihak yang terlibat; (iii) Membuat jejak catatan tertulis.

Undang-Undang Kerahasiaan Bank telah menjadi salah satu alat utama untuk memerangi pencucian uang. Namun, hingga tahun 2012, BSA mensyaratkan Laporan Aktivitas Mencurigakan (SAR) yang diajukan ke Kantor Pengawas Mata Uang (OCC) tidak didigitalisasi dan memiliki komponen pemrosesan manual, yang menjadi kendala, dan ditingkatkan melalui sistem pengarsipan elektronik yang meningkatkan kecepatan dan efisiensi pemrosesan. Kemudian, Undang-Undang Pengendalian Pencucian Uang (1986) menetapkan pencucian uang sebagai kejahatan federal. Undang-Undang Anti-Penyalahgunaan Narkoba tahun 1988 memperluas cakupan dari bank dan lembaga keuangan untuk mencakup real estat, dealer otomotif dalam persyaratan pelaporan transaksi mata uang besar (lebih dari Rp 30 juta).

Undang-Undang Patriot (2001) dan Undang-Undang Reformasi Intelijen Pencegahan Terorisme (2004) memperluas Undang-Undang Kerahasiaan Bank dengan memasukkan pendanaan terorisme sebagai kejahatan, melarang lembaga keuangan untuk terlibat dengan bank cangkang asing, meningkatkan uji tuntas dan prosedur kenali klien Anda, dan akhirnya meningkatkan pertukaran informasi lintas lembaga di antara lembaga pemerintah dan lembaga keuangan. Terakhir, Undang-Undang Reformasi Intelijen Pencegahan Terorisme tahun 2004 memperluas Undang-Undang Kerahasiaan Bank dengan peraturan yang mewajibkan pengiriman dana elektronik lintas batas. Sebagian besar undang-undang ini menyediakan data dan mekanisme akuisisi data yang berharga yang penting agar solusi deteksi dapat berjalan.

Selain hukum kejahatan keuangan, perusahaan dan transaksi keuangan tunduk pada banyak peraturan yang mencakup berbagai aspek mulai dari keadilan dan diskriminasi hingga kualitas data, penjelasan dan interpretasi hingga persyaratan keamanan siber. Sebagai contoh, transaksi kartu kredit, aplikasi pinjaman dan hipotek, dan transaksi keuangan lainnya dapat tunduk pada Undang-Undang Perlindungan Kredit Konsumen, Undang-Undang Pelaporan Kredit yang Adil, Undang-Undang Kesempatan Kredit yang Setara, Undang-Undang Transaksi Kredit yang Adil dan Akurat, serta arahan peraturan seperti Peraturan B. GDPR dan CCPA memerlukan kondisi penjelasan, transparansi, dan interpretasi serta persyaratan terkait data. Meskipun hukum dan peraturan tampaknya tidak kritis dari perspektif pengembangan solusi, peraturan tersebut sangat penting dalam merancang pendekatan solusi. Peraturan tersebut menyoroti persyaratan mendasar (seperti interpretasi, transparansi, dll.), sumber data, persyaratan kualitas, dan batasan operasional.

Teknik Deteksi Berbasis Grafik

AI dan Pembelajaran Mesin Tradisional

Pembelajaran mesin telah digunakan dalam deteksi penipuan sejak tahun 1990-an. Selama bertahun-tahun, teknik pemeriksaan grafik dan jaringan telah ditetapkan sebagai alat penting baik dalam pembelajaran maupun praktik industri. Grafik secara inheren menunjukkan keunggulan dalam merepresentasikan data transaksi keuangan yang mendasarinya. Node dan edge sering kali merepresentasikan perusahaan, individu, rekening, transfer dana, lokasi, perangkat, dan data keuangan atau non-keuangan lainnya. Tergantung pada aplikasinya, beragam jenis grafik (terarah, tidak terarah, siklik, asiklik, statis, dinamis, beratribut, dan berwarna) telah dimanfaatkan secara efektif.

Penambangan Data

Teknik penambangan data awal untuk deteksi penipuan bergantung pada jaringan saraf, regresi, mesin vektor pendukung, jaringan Bayesian yang beroperasi pada representasi tabular. Kemudian, representasi berbasis grafik telah dieksplorasi seperti pemilihan komunitas kepentingan, grafik dinamis dan sistem berbasis tanda tangan.

Deteksi Anomali Graf

Deteksi anomali telah diterapkan dalam sistem layanan keuangan untuk menangkap karakteristik pembeda kecurangan dalam jumlah data transaksi keuangan yang sangat besar. Hal ini memberikan wawasan tentang pola data dengan berfokus pada karakteristik yang

berbeda dalam hal konektivitas, karakteristik entitas, aliran, pola lalu lintas (baik lokal maupun global), karakteristik sub-graf, dan peristiwa dalam representasi graf.

Pemeriksaan Sub-graf dan Terawasi

Pemeriksaan dan penambangan sub-graf memeriksa pola graf lokal melalui pembelajaran terawasi serta semi atau tidak terawasi. Selain konektivitas, teknik aliran jaringan tradisional (misalnya min-cut/max-flow) membantu mengidentifikasi sub-graf yang diminati. Sebagai contoh, pemeriksaan sub-graf dapat mengidentifikasi jaringan kecurangan skala kecil yang menghasilkan pola abnormal pada sejumlah besar akun. Namun demikian, karena pola kecurangan sangat dinamis dan bersifat antagonis, penggunaan teknik deteksi kecurangan menyebabkan perubahan taktik kecurangan untuk mencegah deteksi. Misalnya, penggunaan komponen yang terhubung kuat oleh algoritma deteksi berbasis grafik telah memotivasi pelaku untuk menyembunyikan aktivitas mereka dengan menciptakan jaringan secara artifisial untuk menyamarkan aktivitas mereka. Penggunaan pemeriksaan berbasis metrik untuk mendeteksi kasus-kasus tersebut telah dieksplorasi oleh beberapa peneliti.

Pemeriksaan Aliran dan Jalur

Dalam pencucian uang, proses pelapisan mengarahkan aliran dana ilegal melalui banyak pihak untuk mencegah deteksi. Salah satu keterbatasan teknik berbasis sub-grafik padat adalah bahwa teknik tersebut sebagian besar berfokus pada transfer satu langkah. Oleh karena itu, teknik tersebut menghadapi kesulitan dan memerlukan penyesuaian untuk mendeteksi kasus multi-tahap yang sering digunakan dalam pencucian uang. Baru-baru ini, solusi multi-partit dan multi-langkah, pemeriksaan aliran dan teknik berbasis lingkungan k-langkah telah diusulkan untuk mengatasi hal ini. Solusi aliran jaringan, yang secara tradisional digunakan untuk deteksi intrusi, juga sangat menarik untuk kasus penggunaan deteksi kejahatan keuangan.

Pembelajaran Mesin Berbasis Graf

Dalam beberapa tahun terakhir, ML berbasis graf dan jaringan saraf graf telah mendapatkan minat yang signifikan. Jaringan saraf graf memberikan keuntungan dibandingkan jaringan saraf tradisional dalam memberikan generalisasi yang lebih baik dan penalaran relasional yang lebih baik. Aplikasinya dalam deteksi kejahatan keuangan telah menghasilkan hasil yang menjanjikan. Meskipun masih dalam tahap awal eksplorasi dalam deteksi penipuan, GNN memberikan representasi yang fleksibel baik dalam hal atribut maupun struktur graf. Mereka memungkinkan tingkat konfigurabilitas struktural bersama dengan kemampuan untuk menyusun arsitektur yang berisi banyak blok. Jaringan saraf graf dapat beroperasi pada berbagai jenis graf dan banyak pembelajaran menunjukkan GNN dalam deteksi kejahatan keuangan. Weber menggunakan jaringan konvolusional graf (GCN) untuk mendeteksi pencucian uang. Pareja mengeksplorasi pertimbangan dinamis dalam jaringan graf, yang penting dalam aplikasi penipuan. Don mengusulkan solusi untuk mengungkap penipu yang menyamar melalui GNN. Studi selanjutnya bertujuan untuk meningkatkan keterbatasan struktural jaringan graf melalui mekanisme perhatian untuk efisiensi yang lebih tinggi.

Kasus Penggunaan Grafik

Teknik grafik digunakan dalam berbagai aspek deteksi penipuan.

Penilaian Transaksi dan Peristiwa

Teknik grafik umumnya digunakan dalam penilaian risiko penipuan transaksi dan mendeteksi peristiwa mencurigakan. Penilaian dan deteksi risiko sering dilakukan secara real-time saat transaksi sedang diproses. Hasilnya dibandingkan dengan ambang risiko yang telah ditentukan dan dapat memicu peringatan untuk peristiwa terkait kejahatan keuangan.

Penilaian Entitas

Meskipun tidak begitu dikenal, teknik komputasi grafik juga digunakan untuk pembuatan Daftar Hitam dan Daftar Abu-abu serta pelacakan jaringan penipuan. Dalam aplikasi ini, teknik grafik digunakan untuk menilai skor risiko entitas berdasarkan profil historis dan peristiwa transaksi mereka (termasuk transaksi moneter dan non-moneter seperti akses/login akun, dll.). Entitas yang diberi skor dapat berupa akun, klien, serta entitas seperti perangkat seperti komputer dan telepon seluler, jaringan area lokal, mesin ATM, dll. Entitas yang dianggap berisiko tinggi dapat dimasukkan ke dalam daftar hitam untuk mencegah transaksi lebih lanjut yang melibatkan mereka atau dapat ditambahkan ke daftar pelacakan untuk terus memantau perilaku mereka.

Pemberian Skor Sub-Graf

Selain node individual, kelompok node dan sub-graf dapat diperiksa dan dilacak untuk mengungkap organisasi kejahatan keuangan melalui karakteristik transaksi serta melalui pembuatan profil sub-graf. Mirip dengan pemberian skor entitas, pemberian skor sub-graf seringkali dapat dilakukan secara offline, terlepas dari batasan pemrosesan transaksi waktu nyata yang memengaruhi pemberian skor transaksi.

Kasus Penggunaan Operasional dan Pemrosesan Peringatan

Meskipun tidak umum diketahui, teknik komputasi graf digunakan dalam kasus penggunaan operasional dalam sistem pemrosesan penipuan. Salah satu contoh umum adalah pemrosesan peringatan. Model deteksi penipuan dan kejahatan keuangan seringkali menghasilkan sejumlah besar peringatan penipuan selama operasi normalnya. Peringatan ini diproses melalui berbagai saluran, termasuk analis penipuan dan sistem pemrosesan peringatan otomatis (misalnya pesan teks dan peringatan seluler). Tergantung pada kasus penggunaannya, pemrosesan peringatan dapat terjadi selama atau setelah pemrosesan transaksi.

Untuk saluran dengan throughput tinggi seperti transaksi pembayaran kartu, pemrosesan peringatan terjadi setelah keputusan persetujuan/penolakan dibuat oleh sistem pemrosesan transaksi. Hal ini berbeda dengan saluran lain, seperti transaksi transfer kawat, *Automated Clearing House* (ACH), dan pembayaran online/tagihan, di mana pemrosesan peringatan dapat berkisar dari akhir hari kerja hingga batas waktu transaksi itu sendiri. Hal ini secara signifikan mengubah karakteristik temporal dari tugas pemrosesan. Namun, untuk sebagian besar saluran, karena banyaknya peringatan yang dihasilkan, sistem pemrosesan peringatan manual mungkin menghadapi keterbatasan bandwidth. Memprioritaskan akun

berdasarkan penilaian risiko melalui pemeriksaan grafik memberikan solusi untuk mengurangi kerugian finansial dan meningkatkan waktu respons.

Investigasi Kejahatan

Selama pemrosesan peringatan dan peringatan mencurigakan, pemeriksaan penipuan dan kejahatan keuangan oleh manusia bekerja sama dalam investigasi. Investigasi ditingkatkan melalui visualisasi grafik serta pemeriksaan peristiwa berbasis grafik. Demikian pula, teknik komputasi grafik memberikan peluang untuk meningkatkan peringatan dengan meningkatkan interpretasi. Singkatnya, komputasi grafik menyediakan sejumlah besar peluang solusi selama proses deteksi, investigasi, dan pencegahan kejahatan keuangan dari ujung ke ujung.

13.3 TANTANGAN KOMPUTASI GRAF PADA DETEKSI FRAUD

Bagian ini membahas beberapa pertimbangan implementasi praktis umum dan kendala dari perspektif pengembangan model.

Pertimbangan Temporal

Sebagian besar solusi deteksi kejahatan keuangan dan penipuan diimplementasikan dalam sistem pemrosesan transaksi. Sistem kompleks ini memproses sejumlah besar data transaksi secara real-time. Sistem pemrosesan transaksi biasanya memiliki SLA waktu respons dalam kisaran milidetik, termasuk waktu untuk memproses transaksi itu sendiri (mengakses informasi akun, memeriksa ketersediaan dana), penilaian kecurangan, pemrosesan jaringan pembayaran, protokol komunikasi, dll. Solusi komputasi graf menghadapi tekanan waktu respons yang serius karena kombinasi kendala pemrosesan waktu nyata dan penggunaan graf besar yang saling terhubung. Dalam dekade terakhir, solusi tradisional mengandalkan representasi graf yang efisien memori, teknik kompresi graf dan komputasi graf terdistribusi untuk menangani graf besar dalam kasus penggunaan keuangan.

Ekstraksi Pola Melalui Perhatian

Teknik perhatian telah mendapatkan popularitas yang meningkat dalam beberapa tahun terakhir, karena efektivitas alaminya dalam deteksi penipuan. Pola penipuan seringkali dirancang secara halus untuk mencegah deteksi oleh pihak berwenang. Banyak jaringan penipuan mempraktikkan periode priming akun yang direncanakan dengan baik untuk menyembunyikan pola transaksi dengan pola serupa yang membuatnya tampak lebih normal. Contoh umum dari hal ini dalam kasus ATO adalah mengirimkan sejumlah kecil uang ke akun pelaku yang mungkin tidak memicu peringatan, namun akan membuat transfer yang lebih besar berikutnya tampak normal dengan membangun riwayat. Perhatian grafik telah digunakan untuk mendeteksi pola tersebut dalam pengambilalihan akun.

Penempatan dana selama pencucian uang melibatkan pola transaksi yang sulit dideteksi yang dilakukan oleh banyak entitas. Tujuannya adalah untuk mentransfer dana antara node sumber dan tujuan yang disembunyikan dengan cermat melalui berbagai saluran transaksi. Demikian pula, selama fase penataan, dana dibagi menjadi jumlah yang lebih kecil untuk mencegah pelaporan. Teknik perhatian menjadi penting untuk mendeteksi kasus-kasus tersebut dan mengidentifikasi pola-pola penting meskipun terkubur dalam volume transaksi yang besar. Demikian pula, jaringan perhatian graf menghilangkan kebutuhan untuk

mengetahui struktur graf terlebih dahulu dan memungkinkan fokus pada bagian input yang paling relevan. Aplikasi penipuan sering meminjam teknik dari ruang keamanan siber (seperti deteksi intrusi) sebagai respons terhadap tekanan ketangkasan. Pengoptimalan sistem untuk pembaruan data, pemilihan fitur temporal khusus, dan pembelajaran sensitif biaya telah dipinjam dan dieksplorasi dari solusi deteksi intrusi. Studi lain mengadaptasi model jaringan konvolusional graf (GCN) sepanjang dimensi temporal untuk menangani kompleksitas dinamis.

Graf Dinamis dan Pembaruan Data

Secara global, bank menengah hingga besar melayani puluhan juta klien setiap hari melalui berbagai saluran pemrosesan. Saluran pembayaran saling terhubung melalui entitas (seperti klien dan rekening), sehingga throughput transaksi diterjemahkan menjadi puluhan juta pembaruan pada representasi graf yang mendasarinya. Jenis transaksi frekuensi tinggi (misalnya pembayaran P2P dan pembayaran kartu kredit) memerlukan pembaruan yang lebih sering daripada saluran yang lebih lambat (misalnya pembayaran tagihan). Meskipun masih dalam tahap awal, teknik-teknik terbaru berfokus pada sifat dinamis sistem transaksi melalui jaringan saraf grafik dinamis.

Selama penerapan sistem grafik, baik node (mewakili entitas seperti pemegang rekening) maupun edge (mewakili hubungan seperti transfer dana) dapat diperbarui. Mendeteksi anomali dalam grafik yang terus berubah dengan cepat pada skala industri merupakan masalah pembelajaran yang signifikan. Baru-baru ini, teknik telah diusulkan untuk mendeteksi anomali secara real-time atau mendekati real-time untuk menentukan apakah edge yang masuk bersifat anomali segera setelah diterima. Deteksi anomali untuk grafik dengan karakteristik sementara dapat dilakukan pada tingkat node, pada tingkat edge, serta pada tingkat sub-grafik. Jaringan saraf grafik yang berevolusi, GNN streaming, jaringan temporal juga telah diusulkan untuk melakukan pembelajaran mendalam pada grafik dinamis. Meskipun masih dalam tahap awal, solusi-solusi ini menargetkan berbagai durasi tautan dan frekuensi pembaruan.

Masalah Skala dan Kompleksitas

Sistem pemrosesan transaksi keuangan biasanya kompleks dan melibatkan banyak jenis dan model transaksi pembayaran untuk memproses risiko penipuan dan strategi bisnis.

Jumlah, Ukuran, dan Variasi Grafik

Dalam sistem pembayaran, fitur yang menarik bagi model deteksi penipuan dapat berada di berbagai grafik referensi dengan jenis dan karakteristik yang berbeda (baik internal lembaga keuangan maupun dari sumber eksternal). Kompleksitas ini menyebabkan kesulitan dalam mengembangkan solusi grafik yang efektif. Berkat karakteristik strukturalnya, jaringan saraf grafik memiliki potensi paralelisme, yang secara teori memungkinkan untuk melakukan komputasi batch dari grafik independen. Baru-baru ini, teknik telah diusulkan untuk memberikan kemampuan untuk beroperasi pada berbagai jenis grafik. Demikian pula, penipuan lintas saluran memerlukan komputasi pada beberapa grafik dengan karakteristik yang berbeda secara bersamaan di bawah tekanan perjanjian tingkat layanan (SLA) respons waktu nyata. Meskipun terbatas, beberapa pendekatan solusi telah diusulkan untuk

mengintegrasikan beberapa grafik dalam struktur terpadu dan melakukan propagasi label, namun belum ada solusi yang jelas untuk mengatasi tantangan praktis.

Grafik besar, seperti yang digunakan dalam aplikasi deteksi kejahatan keuangan dan penipuan, menimbulkan tantangan unik karena ukurannya yang sangat besar. Dalam grafik transaksi, puluhan juta pelanggan perlu direpresentasikan dalam sistem pembayaran, dengan ukuran dan kompleksitas interkonektivitas yang serupa. Mengekstraksi wawasan pada grafik besar itu sendiri atau sub-grafik di dalamnya, tetap menjadi tantangan. Merepresentasikan perbedaan karakteristik sub-grafik yang berbeda sangat penting dalam mengidentifikasi kejahatan keuangan. Sebagai contoh, kumpulan node yang tampak sebagai klik dalam grafik dapat mewakili jaringan penipuan pencucian uang yang tersembunyi. Dalam beberapa tahun terakhir, teknik untuk mengkategorikan dan mendeskripsikan perbedaan di berbagai wilayah grafik telah diusulkan. Nilai proposisi teknik tersebut lebih tinggi dalam kasus di mana interpretasi sub-grafik yang berbeda diperlukan untuk tujuan investigasi dan resolusi peringatan pada grafik besar.

Kompleksitas Tingkat Entitas dan Fitur

Data transaksi keuangan biasanya mencakup pihak-pihak yang terlibat dalam transaksi, aliran dana, konektivitas historis antar pihak, peristiwa lintas saluran, profil risiko, dan banyak fitur lainnya. Dalam transaksi pembayaran kartu, ratusan fitur diteruskan dari jaringan pembayaran selama pemrosesan transaksi dan digabungkan dengan sejumlah besar fitur secara internal. Tergantung pada kasus penggunaan, sejumlah besar fitur kompleks dapat digunakan dalam teknik berbasis grafik dan dihitung secara real-time untuk mencapai keputusan persetujuan/penolakan tepat waktu.

Kompleksitas Algoritma dan Waktu Respons

Meskipun tidak ditekankan dalam sebagian besar studi, kompleksitas algoritma juga menarik untuk implementasi skala besar dan sensitif waktu seperti deteksi penipuan. Jaringan saraf grafik telah terbukti efektif pada jaringan yang jauh lebih besar daripada yang telah dilatih, yang merupakan keuntungan dalam implementasi skala besar. Namun, kombinasi ukuran dan kompleksitas algoritma menimbulkan tantangan implementasi yang serius.

Meningkatnya Kompleksitas Penipuan

Seperti yang telah disebutkan di bagian sebelumnya, skema penipuan semakin kompleks selama bertahun-tahun. Kompleksitas ini terlihat hampir di semua dimensi, termasuk jumlah saluran yang terlibat, pola aktivitas dan peristiwa, karakteristik temporal, entitas yang terlibat, dll. Sebagai contoh, penipuan pengambilalihan akun seringkali melibatkan sejumlah besar peristiwa non-moneter yang halus yang terjadi di berbagai saluran, termasuk panggilan telepon ke pusat layanan pelanggan, transaksi non-moneter online (seperti perubahan pengaturan akun), peristiwa akses dan login akun (seperti permintaan kata sandi satu kali (OTP)), transaksi moneter (seperti pembayaran kartu, transfer kawat, transaksi antar individu), dll.

Selama proses deteksi, peristiwa yang terjadi dalam jangka waktu lama (kadang-kadang berbulan-bulan) dievaluasi sehubungan dengan sejumlah besar entitas dan profil risikonya. Tujuannya adalah untuk mengidentifikasi risiko dan titik kompromi akun (seperti malware

pada perangkat yang digunakan untuk login). Kemudian, transaksi, peristiwa, dan entitas (seperti akun, perangkat) perlu ditandai untuk diproses. Bahkan skema penipuan tradisional satu saluran pun secara bertahap bergeser ke peristiwa multi-saluran yang kompleks. Hal ini menciptakan kompleksitas baik dalam algoritma deteksi, jumlah dan kompleksitas grafik serta teknik grafik yang terlibat. Hal ini juga membutuhkan transisi yang mulus antara grafik, platform, sistem pemrosesan transaksi, saluran, dll., serta integrasi untuk mendeteksi pola anomali di berbagai saluran. Tahapan individual kejahatan keuangan mungkin melibatkan sistem transaksi dan pembayaran yang berbeda, membutuhkan strategi dan teknik deteksi yang berbeda, sementara solusi yang terpisah perlu diintegrasikan untuk menyusun pandangan yang koheren dan memeriksa aktivitas mencurigakan di berbagai platform dan sistem.

Kualitas, Kompleksitas, dan Noise Data

Selain noise yang terjadi secara alami dalam data, dalam kasus penggunaan penipuan, para penjahat sering mencoba menyuntikkan data yang menyesatkan ke dalam sistem pemrosesan transaksi keuangan untuk menyembunyikan aktivitas mereka. Dalam penipuan akun sintetis, pengambilalihan akun, dan kejahatan lainnya, data tersebut dapat menjadi faktor penentu dalam proses pengambilan keputusan. Meskipun ada strategi mitigasi untuk serangan tersebut, ketahanan dan kinerja solusi algoritmik yang mendasarinya di bawah tekanan kualitas data jarang dipertimbangkan dalam studi pembelajaran. Solusi deteksi kejahatan keuangan menggunakan beragam jenis data dan sumber data seperti data tidak terstruktur dari gambar, video, rekaman audio, teks, email, data terstruktur dari transaksi keuangan, informasi akun, dan catatan historis. Namun, proses integrasi dan pembaruan data yang berkelanjutan, ditambah dengan masalah kualitas dan kompleksitas data, menciptakan lingkungan yang menuntut dan membutuhkan solusi yang tepat sasaran.

Ketersediaan Data Berlabel dan Dukungan Infrastruktur

Meskipun sebagian besar sistem deteksi berbasis transaksi (seperti pembayaran kartu) memiliki dukungan infrastruktur yang mapan dan otomatis untuk pembuatan data berlabel, kasus penggunaan lain bervariasi dalam hal upaya yang diperlukan untuk menghasilkan data tersebut. Tergantung pada kasus penggunaannya, keterbatasan data berlabel juga dapat disebabkan oleh tingkat kecurangan yang rendah, kurangnya pelaporan, atau integrasi data yang tepat waktu. Meskipun solusi deteksi anomali menunjukkan keunggulan alami dalam kasus-kasus dengan data berlabel yang terbatas, peningkatan kinerja solusi tanpa pengawasan dan semi-pengawasan masih menjadi tantangan.

Standar Algoritma, Tolok Ukur, dan Alat

Seperti yang telah disorot sebelumnya, algoritma graf telah semakin populer di banyak domain aplikasi termasuk layanan keuangan. Upaya standardisasi, tolok ukur, dan alat bermanfaat untuk aplikasi akademis dan sistem penerapan industri. Keragaman yang mendasari blok bangunan primitif algoritma graf dapat menjadi tantangan karena kelompok yang berbeda menciptakan varian tumpang tindih yang berbeda alih-alih mengembangkan blok bangunan tingkat rendah yang umum.

Baru-baru ini, telah ada upaya untuk mengembangkan serangkaian blok bangunan primitif standar untuk teknik graf. Demikian pula, kerangka kerja tolok ukur untuk jaringan saraf graf untuk menambahkan model baru untuk kumpulan data arbitrer telah diusulkan baru-baru ini. Demikian pula, dari perspektif komputasi, kerangka kerja tolok ukur telah dikembangkan untuk meningkatkan efisiensi algoritma graf. Meskipun aplikasi keuangan belum termasuk dalam tolok ukur tersebut, upaya peningkatan berkelanjutan kemungkinan akan mencakup kustomisasi layanan keuangan dalam waktu dekat.

Sifat Adversarial dan Isu Ketahanan

Kejahatan keuangan adalah contoh klasik dari aplikasi adversarial dinamis. Adaptabilitasnya dapat berupa:

(i) *Melewati Langkah-Langkah Pencegahan*: Seperti yang dibahas di Bagian 1, pendekatan deteksi penipuan berbasis aturan sebagian besar menjadi tidak efektif karena pelaku dengan cepat memperkirakan aturan statis dan mengembangkan teknik untuk melewatinya. Teknik pembelajaran terawasi juga menghadapi tantangan serius, karena penipu sering mengadopsi teknik baru untuk menghindari deteksi berbasis pembelajaran mesin. Sebagai contoh, ketika transaksi pembayaran curang dari organisasi kejahatan keuangan ditolak oleh sistem deteksi berbasis pembelajaran mesin, taktik baru sering digunakan segera.

(ii) *Penggunaan Saluran dan Pergeseran Skema Penipuan*: Seperti yang disebutkan sebelumnya, topologi kejahatan keuangan sebagian besar telah bergeser dari kasus saluran tunggal yang lebih sederhana ke kasus multi-saluran yang kompleks. Demikian pula, penjahat telah menggunakan saluran baru yang secara tradisional tidak terkait dengan jenis penipuan yang sesuai. Sebagai contoh, laporan industri mengklaim bahwa saluran-saluran baru seperti transaksi bitcoin semakin populer untuk kegiatan pencucian uang.

(iii) *Taktik yang Berubah Cepat dari Waktu ke Waktu*: Penipuan pembayaran antar individu (P2P) seluler dikenal menunjukkan pergeseran taktik yang tiba-tiba, yang menciptakan lingkungan yang menantang bagi solusi berbasis pembelajaran terawasi. Deteksi anomali, dan algoritma adaptif telah dieksplorasi dan diterapkan untuk kasus penggunaan ini. Dalam jangka panjang, teknik adaptif dan adversarial memberikan jalur solusi yang paling menjanjikan karena mencerminkan lingkungan aplikasi dengan paling akurat.

(iv) *Ketahanan*: Taktik adversarial juga memainkan peran penting dalam ketahanan keseluruhan solusi deteksi. Seringkali, organisasi kriminal memiliki kemampuan untuk bereaksi terhadap tindakan pencegahan dengan cepat dan dalam skala besar. Peningkatan investasi teknologi dan kemampuan otomatisasi mereka menjadi perhatian yang semakin meningkat. Dalam sistem praktis, kemampuan ini dapat memiliki implikasi serius terhadap ketahanan model deteksi berbasis pembelajaran mesin.

Peningkatan Otomatisasi dan Serangan Pembelajaran Mesin

Serangan pembelajaran mesin menjadi perhatian yang semakin meningkat karena menargetkan aplikasi yang sangat penting. Banyak pembelajaran menunjukkan bahwa model ML mungkin menghadapi masalah ketahanan selama kemungkinan serangan peracunan, penghindaran, dan inferensi. Jaringan saraf dikenal sensitif terhadap gangguan kecil sekalipun pada data dan karenanya rentan terhadap serangan adversarial. Membangun model yang tangguh telah menjadi tujuan implementasi utama dalam kasus penggunaan industri. Membangun strategi melawan serangan peracunan dan jenis serangan pembelajaran mesin lainnya telah mendapatkan minat yang signifikan dalam beberapa tahun terakhir. Pada titik ini, pembelajaran tentang ketahanan dan tantangan adversarial untuk sistem pembelajaran mesin berbasis grafik masih dalam tahap awal.

Fakta bahwa jaringan penipuan semakin bergantung pada otomatisasi dan teknologi canggih, menyoroti batasan mendasar keberhasilan teknik pencegahan penipuan saat ini dan generasi berikutnya. Kelincahan, otomatisasi, dan skala menjadi sangat penting dalam sistem penerapan. Solusi komputasi graf perlu beradaptasi dengan cepat terhadap tantangan yang semakin meningkat terkait ketahanan dan serangan pembelajaran mesin.

Tata Kelola dan Kepatuhan Regulasi

Meskipun sebagian besar hukum dan peraturan kejahatan keuangan melibatkan prosedur dokumentasi dan pelaporan, proses tata kelola model di perusahaan keuangan menghadirkan lingkungan yang menantang bagi pembelajaran mesin dan teknik berbasis grafik. Lembaga keuangan seringkali memerlukan pendekatan pemodelan konservatif terhadap jaminan kinerja model untuk tujuan kepatuhan. Akibatnya, banyak teknik yang menjanjikan mengalami masalah selama pemantauan tata kelola model (karena risiko inherennya dan kurangnya jaminan kinerja untuk kepatuhan).

Di luar tantangan persetujuan, prosedur tata kelola model dan manajemen risiko membutuhkan waktu yang jauh lebih lama untuk teknik baru. Hal ini memperpanjang waktu penerapan dan membahayakan jadwal pengembangan model dan kinerja deteksi. Sebagai contoh, pembelajaran penguatan memberikan keuntungan utama dalam lingkungan yang bersifat antagonis berkat asal-usulnya dalam permainan. Studi pembelajaran penguatan grafik telah menunjukkan hasil yang menjanjikan dalam kasus penggunaan deteksi kejahatan keuangan. Namun, implementasinya di industri jasa keuangan lebih lambat. Demikian pula, pendekatan pembelajaran otonom dan algoritma online sangat menarik untuk deteksi penipuan. Meskipun demikian, otonomi yang mendasari dan kurangnya jaminan kinerja menjadi masalah bagi pemantauan kepatuhan peraturan internal. Proses tata kelola model saat ini memerlukan revisi dan penyesuaian untuk mempercepat adopsi pendekatan pembelajaran mesin yang lebih otonom untuk deteksi penipuan.

Interpretasi dan Penjelasan

Peraturan terbaru seperti CCPA dan GDPR memberlakukan mandat interpretasi dan penjelasan pada sebagian besar model AI/ML di layanan keuangan. Namun, memperluas kemampuan solusi penjelasan saat ini telah menjadi tantangan di seluruh industri. Penjelasan dan interpretasi menjadi lebih menantang dalam grafik dinamis besar yang digunakan dalam

aplikasi deteksi kejahatan keuangan. Di luar persyaratan peraturan, teknik penjelasan diperlukan untuk tujuan audit dan penyelidikan untuk transaksi yang ditolak serta untuk investigasi kejahatan. Baru-baru ini, beberapa pembelajaran telah mengeksplorasi tantangan ini dalam grafik dinamis. Mengembangkan solusi penjelasan yang dioptimalkan untuk model grafik yang muncul dan memenuhi persyaratan peraturan sangat penting.

Visualisasi dan Investigasi Kejahatan

Visualisasi digunakan untuk berbagai tujuan dalam sistem deteksi kejahatan, termasuk namun tidak terbatas pada pelacakan tren penipuan yang muncul, melakukan investigasi kriminal, menyelidiki peringatan penipuan, dan berkolaborasi dengan analis penipuan lainnya. Meskipun telah digunakan secara efektif di bidang lain, visualisasi konvensional dan alat human-in-the-loop menghadapi hambatan serius dalam deteksi penipuan. Komputasi graf memberikan keuntungan intrinsik dalam merepresentasikan dan memvisualisasikan data. Namun, penskalaan solusi ke skala industri dan penanganan tantangan kecepatan dan kompleksitas merupakan masalah serius. Dalam beberapa waktu terakhir, solusi visualisasi graf besar telah diuji coba pada kisaran puluhan ribu node. Kemudian visualisasi super-graf diusulkan untuk ratusan ribu node. Namun, visualisasi transaksi keuangan membutuhkan visualisasi graf untuk graf dalam ratusan juta, yang masih menimbulkan tantangan serius. Demikian pula, persyaratan interaktivitas, kompleksitas graf, jumlah graf yang terlibat, dan masalah interpretasi menghambat solusi komputasi graf dan visualisasi yang terkait.

Mengintegrasikan Teknik dan Sumber Data yang Baru Muncul

Pendekatan baru yang melibatkan pembuatan profil perilaku dan peristiwa non-moneter memberikan hasil yang menjanjikan. Solusi ini bergantung pada sejumlah besar data dari berbagai sumber data hibrida, seperti aliran klik, login, tampilan halaman, profil perangkat seluler, dll. Mengintegrasikan solusi tersebut dan sumber data hibrida dengan representasi berbasis grafik tradisional dan tipe data dalam sistem pemrosesan transaksi yang kompleks merupakan tantangan yang sedang berkembang.

13.4 KESIMPULAN DAN TANTANGAN DETEKSI FRAUD BERBASIS GRAF

Dalam artikel ini, kami meninjau tantangan aplikasi umum dalam sistem deteksi penipuan dan kejahatan keuangan berbasis grafik. Kejahatan keuangan dan skema penipuan telah menunjukkan perubahan pesat dalam beberapa tahun terakhir. Didorong oleh pertumbuhan lanskap pembayaran digital, topologi dan karakteristik kejahatan (dari jenis transaksi hingga saluran dan alat) telah berubah, yang membuat solusi deteksi tradisional menjadi tidak efektif. Bahkan teknik pembelajaran mesin pun menghadapi masalah dalam menangani pola penipuan yang semakin halus. Teknik grafik memberikan keuntungan intrinsik dalam mendeteksi penipuan dalam volume besar data transaksi digital. Baik solusi komputasi grafik tradisional maupun teknik yang muncul sangat menarik dari perspektif penerapan industri. Namun, implementasi dan penerapan teknik komputasi grafik dalam sistem deteksi kehidupan nyata menimbulkan kesulitan unik, karena ukuran, kecepatan, kompleksitas, dan karakteristik yang merugikan dari aplikasi deteksi kejahatan keuangan. Selain itu, kompleksitas yang mendasari sistem pemrosesan transaksi digital, termasuk persyaratan implementasi skala

besar, pemrosesan waktu nyata, sifat silo dari saluran, pembaruan yang sering, data/grafik yang kompleks, membuat penerapan dan pencapaian target kinerja deteksi menjadi sulit. Meskipun jaringan saraf graf dan solusi adaptif yang muncul memberikan alat penting untuk membentuk masa depan deteksi penipuan dan kejahatan keuangan, sifat penipuan yang sangat adaptif kemungkinan akan terus menimbulkan tantangan. Seperti yang disoroti di seluruh artikel ini, konvergensi kejahatan keuangan, penipuan, dan ancaman keamanan siber merupakan perhatian signifikan bagi industri jasa keuangan karena sifat sistem yang terpisah dan berbeda. Solusi dan sistem berbasis graf sangat penting untuk menyesuaikan dan mengintegrasikan berbagai sistem deteksi dan pencegahan. Artikel ini berpendapat bahwa fokus pada persyaratan spesifik aplikasi memberikan peluang untuk meningkatkan solusi berbasis graf yang ada dan yang sedang berkembang.

BAB 14
PEMBUATAN JUDUL OTOMATIS BERBASIS TRANSFORMER PRETRAINED

Menciptakan rasa ingin tahu dan minat terhadap suatu topik dalam pembelajaran daring merupakan tugas yang menantang. Pratinjau yang baik yang menguraikan isi jalur pembelajaran dapat membantu peserta didik mengetahui topik tersebut dan tertarik padanya. Untuk tujuan ini, kami mengusulkan pendekatan pembuatan judul hierarkis untuk menghasilkan judul yang relevan secara semantik untuk sumber daya pembelajaran dalam jalur pembelajaran dan judul untuk jalur itu sendiri. Pendekatan kami untuk Generasi Judul Otomatis untuk teks tertentu didasarkan pada Model Bahasa Transformer GPT-2 yang telah dilatih sebelumnya. Sekumpulan judul kandidat dihasilkan dan judul yang sesuai dipilih di antaranya, yang kemudian disempurnakan atau dihilangkan noise-nya untuk mendapatkan judul akhir. Model ini dilatih pada abstrak artikel pembelajaran dari arXiv dan dievaluasi pada tiga set uji yang berbeda. Kami menunjukkan bahwa model ini menghasilkan judul yang relevan secara semantik dan sintaksis seperti yang tercermin dalam skor ROUGE, BLEU, dan evaluasi manusia. Kami mengusulkan modul Summarizer abstrak opsional berdasarkan model Transformer T5 yang telah dilatih sebelumnya untuk mempersingkat dokumen berukuran sedang. Modul ini juga dilatih dan dievaluasi pada artikel pembelajaran dari dataset arXiv. Terakhir, kami menunjukkan bahwa model generasi judul hierarkis yang diusulkan untuk jalur pembelajaran memiliki hasil yang menjanjikan.

14.1 PEMBUATAN JUDUL OTOMATIS UNTUK SUMBER BELAJAR DARING

Dalam beberapa tahun terakhir, kita telah menyaksikan peningkatan yang stabil dalam jumlah pendidik dan peserta didik yang beralih ke pendidikan daring, yang kini meningkat berlipat ganda selama masa pandemi ini. Kita semua tahu bahwa pendidikan daring membawa sejumlah tantangan, salah satunya adalah, menciptakan minat atau motivasi bagi peserta didik untuk tertarik pada pendidikan daring. Kami membayangkan bahwa pratinjau singkat yang menampilkan konten utama dari sumber belajar yang akan dipelajari peserta didik untuk mempelajari suatu mata pelajaran dapat menarik perhatian peserta didik dan menginspirasi mereka untuk benar-benar mulai mempelajari topik tersebut.

Meskipun pembuatan konten seperti itu secara manual dimungkinkan dan dilakukan di beberapa Kursus Daring Terbuka Massif (MOOC) terkenal dan platform pendidikan lainnya, sejauh pengetahuan kami, belum banyak kontribusi pembelajaran dalam menghasilkan konten seperti itu secara otomatis. Pembuatan pratinjau secara manual membutuhkan waktu, upaya, dan anggaran yang signifikan. Hal ini juga membutuhkan ahli materi pelajaran untuk membantu editor dalam menyusun konten. Alasan-alasan ini dapat mempersulit pembuatan pratinjau tersebut dalam skala besar, terutama untuk topik pembelajaran yang baru diperkenalkan atau domain khusus.

Selain itu, di bidang di mana banyak sekali sumber belajar terbuka ditambahkan setiap hari, pendekatan ini menjadi sangat membantu, terutama untuk sumber daya yang tidak dianggap sebagai sumber daya pendidikan, tetapi tetap dianggap bermanfaat. Untuk mengatasi hal ini, kami mengusulkan metodologi pembuatan judul hierarkis secara otomatis untuk satu set atau rangkaian sumber belajar tanpa memerlukan metadata eksplisit atau format standar. Hal ini menciptakan pratinjau sederhana untuk serangkaian sumber belajar yang disebut sebagai "jalur belajar". Jika kita diberikan sekumpulan sumber belajar terbuka dan bukan suatu rangkaian, maka metode yang diusulkan dapat digunakan untuk menghasilkan jalur belajar secara otomatis. Sumber belajar terbuka mengacu pada sumber belajar yang tersedia secara terbuka daring dan dapat dibuat oleh pengguna yang berbeda dan tidak harus memiliki format atau metadata standar. Untuk menghasilkan pratinjau untuk jalur belajar tersebut, pertama, judul dihasilkan untuk setiap sumber belajar dalam jalur tersebut dan kemudian judul dihasilkan untuk jalur itu sendiri. Judul yang dihasilkan untuk sumber daya dan jalur secara keseluruhan membentuk metadata yang dapat berguna untuk pengindeksan dan pengambilan informasi.

Untuk pembuatan judul, kami mengusulkan pendekatan baru yang terdiri dari alur kerja tiga modul yaitu Generasi, Seleksi, dan Penyempurnaan diikuti oleh fungsi Penilaian. Modul Generasi dan Penyempurnaan didasarkan pada model bahasa *Generative Pre-trained Transformer 2* (GPT-2), sedangkan modul Seleksi memiliki pendekatan berbasis heuristik. Terdapat modul Ringkasan opsional berdasarkan Model *Text-to-Text Transfer Transformer* (T5) yang telah dilatih sebelumnya dapat diterapkan untuk sumber daya pembelajaran atau dokumen dengan panjang sedang. Model ini mampu menghasilkan judul yang akurat meskipun memiliki korpus data pelatihan yang relevan lebih kecil karena kemampuan pembangkitan bahasa alami berasal dari pelatihan sebelumnya, sementara model harus mempelajari nuansa spesifik tugas dan korpus.

Seperti yang disebutkan sebelumnya, pembangkitan judul dibangun menggunakan model GPT-2 yang telah dilatih sebelumnya. GPT-2 adalah model bahasa berbasis transformer besar yang dilatih pada 40 GB teks internet, dengan tujuan sederhana untuk memprediksi kata berikutnya berdasarkan semua kata sebelumnya dalam urutan tersebut. GPT-2 menghasilkan sampel teks sintetis yang tampak realistis dan merupakan kelanjutan yang koheren dari input yang diberikan. Namun, model generatif seperti GPT-2 bersifat probabilistik. Oleh karena itu, ketika digunakan untuk tugas tertentu, teks yang dihasilkan dapat berbeda setiap kali model diinferensikan. Dalam kasus kami, model dapat menghasilkan judul yang berbeda setiap kali model diinferensikan. Meskipun ini memiliki keuntungan menghasilkan tidak hanya satu tetapi banyak judul yang akurat secara semantik. Dengan judul-judul yang menarik, menjadi penting untuk memilih yang terbaik di antara judul-judul tersebut.

Untuk itu, kami mengusulkan metode seleksi berbasis aturan yang memilih judul yang paling sesuai di antara judul-judul yang dihasilkan. Ini diikuti oleh model berbasis GPT-2 lainnya yang menyempurnakan atau menghilangkan noise pada judul yang dipilih. Sebagai langkah terakhir, kami menghasilkan skor relevansi (RS) untuk judul yang dipilih dan judul yang disempurnakan, dan judul dengan skor relevansi tertinggi dipilih sebagai judul keluaran akhir.

Dalam kasus aplikasi dengan kemungkinan intervensi pengguna, kami menyajikan judul yang dipilih dan yang disempurnakan kepada pengguna akhir dengan skor relevansi yang disarankan.

Kami menggunakan dataset abstrak artikel pembelajaran dari arXiv untuk melatih model Generasi Judul kami dan kemudian mengevaluasi model ini pada tiga set uji yang berbeda. Eksperimen menggunakan metodologi kami menunjukkan hasil yang menjanjikan ketika dievaluasi menggunakan metrik ROUGE dan BLEU terhadap set uji dan dua dataset yang belum pernah dilihat sebelumnya dari domain yang sama. Hal yang sama berlaku ketika evaluasi manusia dilakukan. Ini menunjukkan bahwa model mudah ditransfer ke dataset apa pun dalam domain minat yang sama. Oleh karena itu, ketika kami menerapkan metodologi generasi judul kami ke sumber daya pembelajaran dari domain yang sama, kami mendapatkan judul yang akurat secara semantik.

Seringkali, sumber belajar (yang direpresentasikan dalam bentuk teks atau transkrip yang dihasilkan untuk sumber belajar) lebih panjang dibandingkan dengan abstrak artikel pembelajaran, oleh karena itu kami mengusulkan modul baru untuk meringkas dokumen tersebut sebelum menerapkan pendekatan pembuatan judul kami. Modul Peringkas ini bersifat opsional dan didasarkan pada T5 yang telah dilatih sebelumnya.

T5 adalah model berbasis *transformer sequence-to-sequence* yang memodelkan tugas Pemrosesan Bahasa Alami (NLP) dalam format teks-ke-teks terpadu dengan input dan output berupa string. Kami melakukan fine-tuning model T5 menggunakan dataset artikel pembelajaran dari arXiv. Kami melakukan berbagai eksperimen dan menghasilkan berbagai model fine-tuned berdasarkan panjang teks input, jenis teks input, dll., dan membandingkan setiap model serta memilih model terbaik di antaranya berdasarkan metrik ROUGE.

14.2 METODE PEMBUATAN JUDUL OTOMATIS

Pembuatan judul atau headline otomatis untuk teks tertentu merupakan masalah yang menarik dan menantang yang telah diteliti oleh berbagai peneliti selama bertahun-tahun. Kami mengeksplorasi dan melaporkan beberapa karya perintis di bidang ini. C´edric membahas pendekatan di mana beberapa kalimat kandidat penting untuk pembuatan judul dipilih. Dari kalimat-kalimat kandidat ini, Frasa Kata Benda (NP) penting diekstrak dan diberi peringkat untuk membuat judul. Jin dan Hauptmann menjelaskan pendekatan untuk pertama-tama mengidentifikasi konten kunci dokumen dalam bentuk konstituen seperti kata atau frasa dan kemudian mengorganisasikannya menjadi kalimat gramatikal menggunakan model bahasa statistik.

Tan mengusulkan pendekatan kasar-ke-halus, yang pertama-tama mengidentifikasi kalimat-kalimat penting dari sebuah dokumen dan kemudian memanfaatkan model peringkasan multi-kalimat dengan perhatian hierarkis untuk memanfaatkan kalimat-kalimat penting untuk pembuatan headline. Colmenares mengusulkan teknik prediksi urutan yang memodelkan pembuatan headline sebagai tugas optimasi diskrit dalam ruang kaya fitur. Putra dan Khodra mengusulkan metode pembuatan judul yang mempertimbangkan niat penulis. Metode ini menggunakan dua pendekatan pembuatan judul, yaitu berbasis templat dan K-

nearest neighbor adaptif. Pendekatan menggunakan arsitektur encoder–decoder atau Transformer. Gu mengusulkan model untuk menghasilkan headline dari sekelompok artikel berita. Mereka menggunakan arsitektur encoder–decoder untuk mengekstrak judul untuk satu artikel, diikuti oleh lapisan perhatian artikel berbasis pemungutan suara mandiri untuk mengekstrak informasi penting yang dimiliki oleh beberapa artikel.

Murao mengusulkan sistem untuk menghasilkan judul dengan dukungan pengeditan dengan melatih model encoder–decoder dari lead dan headline untuk menghasilkan judul pendek. Lopyrev menggunakan arsitektur jaringan saraf berulang encoder–decoder dengan sel unit LSTM dan perhatian untuk menghasilkan judul berita dari artikel berita masukan. Gavrilov menggunakan arsitektur Universal Transformer dengan teknik pengkodean pasangan byte yang menggunakan perhatian diri untuk menghasilkan judul berita dari artikel berita masukan. Arsitektur encoder–decoder atau Transformer yang disebutkan di atas menghasilkan judul yang cukup relevan dan akurat secara semantik tetapi membutuhkan sejumlah besar data pelatihan. Sedangkan, model kami dapat dilatih menggunakan kumpulan data yang lebih kecil yang hanya terdiri dari beberapa ribu titik data.

Kontribusi utama lain dari pendekatan kami adalah menghasilkan kumpulan judul kandidat dan menyediakan metode untuk memilih yang terbaik di antaranya. Ini selanjutnya disempurnakan untuk membuat judul yang akurat secara sintaksis dan semantik, serta representatif. Opsi untuk memilih judul yang dipilih atau judul yang disempurnakan juga disajikan kepada pengguna bersama dengan skor relevansi ketika model digunakan dalam aplikasi yang memungkinkan intervensi pengguna.

Selain metodologi, eksperimen kami juga berfokus pada domain ilmiah dan teknis yang memiliki pola yang menantang dan sejumlah besar konsep unik, tidak seperti dataset berita yang digunakan dalam pendekatan di atas di mana pola struktural seperti garis dasar kalimat utama memiliki pengaruh kuat dalam proses pembuatan judul. Pendekatan kami untuk menghasilkan ringkasan abstrak dari dokumen berukuran sedang didasarkan pada model berbasis transformer T5. Beberapa pendekatan mempelajari pembuatan ringkasan abstrak menggunakan GPT-2 yang telah dilatih sebelumnya, namun keuntungan menggunakan T5 dibandingkan GPT-2 untuk peringkasan adalah bahwa T5 telah dilatih sebelumnya pada tugas peringkasan dan oleh karena itu model tersebut mengetahui bobot peringkasan. Hal ini terbukti dalam hasil model kami yang berkinerja lebih baik jika dibandingkan dengan model peringkasan yang dibangun menggunakan GPT-2.

Kami menerapkan metodologi pembuatan judul kami pada sumber daya pembelajaran dalam jalur pembelajaran dan membuat judul untuk seluruh jalur pembelajaran. Hal ini memberikan gambaran sekilas tentang isi jalur pembelajaran, sehingga mendorong pembelajaran. Belum banyak pekerjaan aktif di bidang ini dan sebagian besar jalur pembelajaran bergantung pada pembuatan judul secara manual. Sejauh pengetahuan kami, satu-satunya karya yang paling mendekati pendekatan kami adalah pembuatan judul hierarkis seperti yang diusulkan oleh Mahapatra. Namun, penulis menghasilkan judul untuk video pendidikan menggunakan segmentasi, sedangkan pendekatan kami dalam menghasilkan judul dapat diterapkan untuk konten tekstual apa pun (bahkan transkrip yang diperoleh dari video).

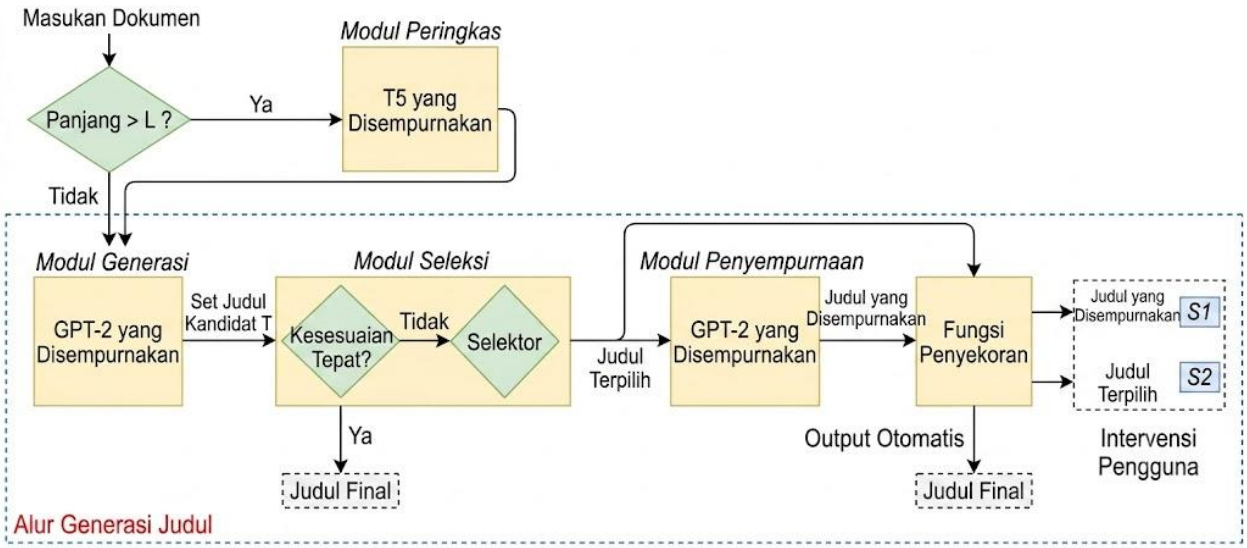
Pendekatan kami juga didasarkan pada pembuatan judul abstrak sedangkan model yang diusulkan bersifat ekstraktif.

14.3 PENDEKATAN GENERASI JUDUL DENGAN FINE-TUNING GPT-2

Pendekatan kami untuk menghasilkan judul untuk dokumen pendek didasarkan pada penyempurnaan model bahasa pra-terlatih Open AI GPT-2. Penyempurnaan model pra-terlatih terbukti sangat meningkatkan kinerja tugas hilir apa pun. Oleh karena itu, kami memilih untuk menggunakan metode ini dalam pendekatan kami. Model pembuatan judul kami terdiri dari tiga modul: Generasi, Seleksi, dan Penyempurnaan, diikuti oleh fungsi Penilaian. Gambar 14.1 menunjukkan alur kerja pendekatan pembuatan judul kami. Abstrak dimasukkan ke modul Generasi, yang menghasilkan kumpulan judul. Dari kumpulan ini, kami mengambil sampel sekumpulan judul kandidat "T" menggunakan strategi pengambilan sampel Top-k, Top-p yang kemudian diteruskan ke modul Seleksi. Dalam modul Seleksi, pendekatan berbasis heuristik digunakan untuk memilih judul terbaik di antara judul-judul yang diambil sampelnya dari kumpulan "T".

Terakhir, judul yang dipilih diteruskan melalui modul Penyempurnaan yang menghilangkan noise pada judul dengan menghasilkan judul yang disempurnakan yang secara sintaksis dan semantik lebih baik. Setelah menghasilkan judul yang disempurnakan, fungsi Penilaian kemudian memberi skor pada judul dari modul Seleksi dan modul Penyempurnaan berdasarkan kesamaan Jaccard. Hal ini membuat model kami tangguh dalam memilih judul akhir dengan memperhatikan yang terbaik dari kedua dunia antara judul yang terdiri dari istilah-istilah yang baru dihasilkan, versus judul yang berisi elemen-elemen konten aktual dalam abstrak.

Terdapat modul Summarizer opsional sebagai pendahulu model Pembuatan Judul kami yang mengubah dokumen berukuran sedang menjadi teks pendek atau abstrak yang bermakna dan ringkas. Modul ini dipanggil ketika panjang dokumen lebih besar dari panjang ambang batas L seperti yang terlihat pada Gambar 14.1. Seperti yang disebutkan sebelumnya, model dapat menyajikan kedua opsi kepada pengguna beserta skor relevansi, dan pengguna dapat memilih judul sesuai kebutuhan mereka dengan memberikan bobot preferensi pada setiap faktor tersebut. Dalam alur kerja otomatis, judul dengan skor relevansi yang lebih tinggi dipilih sebagai judul akhir.



Gambar 14.1 Alur kerja pendekatan pembuatan judul kami.

Modul Generasi

Kami menggunakan model GPT-2 berukuran sedang (345 juta parameter) sebagai pilihan model bahasa pra-terlatih kami. Model bahasa seperti GPT-2 dapat digunakan dalam kerangka kerja teks-ke-teks hanya dengan menggabungkan input dan target dengan menambahkan kata pemisah di antara keduanya. Kami menggunakan pemisah, $TL;\ DR$: yang digunakan untuk peringkasan dalam artikel GPT-2 oleh Radford, karena pembuatan judul adalah sub-tugas dari peringkasan dan kami percaya bahwa GPT-2 pra-terlatih telah mempelajari perilaku peringkasan untuk kata kunci ini.

Secara formal, pertimbangkan setiap titik data $d_i(x_i, y_i)$ dari dataset D_{title} (X, Y) dengan ``n" titik data di mana $x_i \in X$ mewakili abstrak dan $y_i \in Y$ mewakili judul. Misalkan $x_i = w_{x1}, w_{x2}, \dots, w_{xa}$ dan $y_i = w_{y1}, w_{y2}, \dots, w_{yb}$ di mana a dan b adalah panjang x_i dan y_i, masing-masing. Kita mendefinisikan ulang setiap titik data dengan menggabungkannya dengan kata kunci pemisah $t = TL;\ DR$: dan token akhir $e = \langle|endoftext|\rangle$ sehingga titik data sekarang menjadi $d_i = x_i, t, y_i, e$. Untuk kesederhanaan dalam perumusan masalah, kita mengganti setiap kata/simbol dalam titik data dengan u_i. Dengan demikian, titik data menjadi $d_i = u_1, u_2, \dots, u_{L_t}$ di mana panjang titik data $L_t = a + b + 2$ dan $L_t \leq 1024$.

Kita memaksimalkan tujuan seperti yang didefinisikan dalam persamaan berikut:

$$P(D_{\text{title}}) = \prod_{i=1}^{n} \prod_{j=1}^{L_t} P\left(u_j \middle| u_1, u_2, \dots, u_{j-1}; \theta\right). \tag{14.1}$$

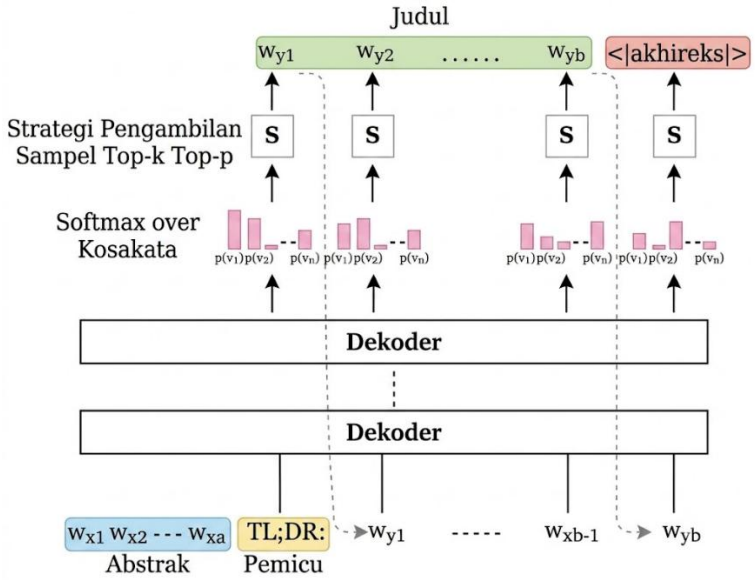
Di sini, $P\left(u_j \middle| u_1, \dots, u_{j-1}; \theta\right)$ adalah probabilitas bersyarat untuk mendapatkan u_j dengan mempertimbangkan token sebelumnya $u_1, \dots, u_{j-1}, \theta$ mewakili parameter model. Kami melakukan fine-tuning pada model bahasa GPT-2 dengan mengoptimalkan kerugian cross-entropy seperti yang didefinisikan dalam persamaan berikut:

$$\text{Loss} = -\sum_i p_i \log_2 p_i' , \tag{14.2}$$

di mana p_i' adalah probabilitas yang dihasilkan oleh model kami untuk kata i dan p_i adalah probabilitas kebenaran dasar dari kata i.

Untuk tujuan dekoding, kami mempertimbangkan berbagai teknik dekoding atau pengambilan sampel seperti Greedy, Beam Search, Top-k dan pengambilan sampel Top-p (Nucleus). Kami tahu bahwa menggunakan metode Greedy untuk mengambil sampel kata menghasilkan dekoding deterministik yang tidak diinginkan mengingat aplikasi kami. Holtzman membahas kerugian menggunakan teknik pengambilan sampel Beam Search dan Top-k. Kelemahan Beam Search adalah tidak cukup beragam dan menghasilkan pola yang berulang. Kelemahan menggunakan pengambilan sampel Top-k adalah menghasilkan sejumlah besar prediksi probabilitas rendah jika distribusinya berpuncak. Kami melihat bahwa teknik pengambilan sampel Top-p menghindari degenerasi teks dengan memotong ekor distribusi probabilitas yang tidak dapat diandalkan, namun, ia menciptakan ruang sampel yang besar ketika distribusinya datar.

Dengan mempertimbangkan pro dan kontra dari kedua kasus tersebut, kami menerapkan versi campuran dari kedua teknik yang disebut pengambilan sampel Top-k, Top-p. Pertama, kami menggunakan pengambilan sampel Top-k di mana kami mengambil sampel k istilah dengan probabilitas softmax tinggi. Kemudian pada himpunan ini, kami menggunakan pengambilan sampel Top-p untuk mendapatkan himpunan kata kunci yang massa probabilitas kumulatifnya setidaknya p. (Nilai k dan p ditentukan secara empiris) Pada himpunan ini, kami mengambil sampel kata berdasarkan distribusi probabilitas bersyaratnya. Pengambilan sampel Top-k, Top-p dengan demikian memastikan bahwa sampel dibatasi dengan k istilah maksimum untuk distribusi datar dan massa probabilitas kumulatif minimum sama dengan p untuk distribusi berpuncak.



Gambar 14.2 Ilustrasi modul Generasi pada waktu inferensi.

Selama fase inferensi, kami menghasilkan himpunan judul kandidat $T = \langle t_1, t_2, \dots, t_n \rangle$ di mana setiap judul t_i mewakili judul yang dihasilkan dari modul Generasi. Gambar 14.2 mengilustrasikan modul Generasi pada waktu inferensi. Seperti yang terlihat pada gambar, kami mulai menghasilkan kata-kata judul $w_{y1}, w_{y2},$. Setelah menambahkan kata kunci $t = TL;\ DR$: di akhir abstrak w_{x1}, w_{x2}, dan berhenti ketika token akhir $\langle |endoftext| \rangle$ dihasilkan oleh model kami. Pada setiap langkah waktu generasi, kami mengambil sampel kata menggunakan strategi pengambilan sampel Top-k, Top-p pada distribusi probabilitas softmax di atas kosakata. Kami juga menerapkan langkah pasca-pemrosesan untuk mengatasi masalah duplikasi kata yang berdekatan agar sintaksisnya tepat.

Modul Seleksi

Teknik pengambilan sampel Top-k, Top-p diharapkan menghasilkan beragam judul ketika diambil sampelnya karena keacakan yang diwarisi pada setiap langkah waktu. Judul yang dihasilkan dari model GPT-2 pada setiap inferensi akan berbeda. Ini berarti bahwa beberapa judul mungkin lebih baik daripada yang lain dalam hal konstruksi linguistik dan semantik untuk merepresentasikan dokumen masukan. Oleh karena itu, kami melakukan inferensi model GPT-2 sebanyak "h" kali untuk abstrak input yang sama untuk mendapatkan judul-judul yang membentuk daftar kandidat T. Kami bereksperimen dengan nilai h yang berbeda, dan berdasarkan deviasi standar skor ROUGE, kami memilih nilai $h = 5$ untuk korpus.

Untuk memilih judul yang paling tepat untuk input yang diberikan dari kumpulan judul kandidat, kami memperkenalkan modul Seleksi dalam pipeline. Kami mengimplementasikan tiga komponen dalam modul Seleksi, yaitu Pencocokan Tepat, Tumpang Tindih NP, dan Kesamaan Semantik. Untuk memilih komponen yang tepat untuk modul ini, kami menggunakan prosedur seleksi maju yang merupakan teknik populer untuk pemilihan fitur dalam pembelajaran mesin.

Kami mulai dengan memilih komponen secara acak. Kemudian kami terus menambahkan masing-masing dari tiga komponen ke modul Seleksi satu per satu, berdasarkan kinerjanya. Tabel 14.1 menunjukkan skor ROUGE-1, 2, L yang diamati pada set validasi saat memilih komponen selama teknik seleksi maju. Angka yang dilaporkan adalah skor rata-rata selama lima kali percobaan. Komponen yang dipilih kemudian membentuk pipeline seleksi serakah tiga langkah seperti yang dijelaskan dalam Gambar 14.3.

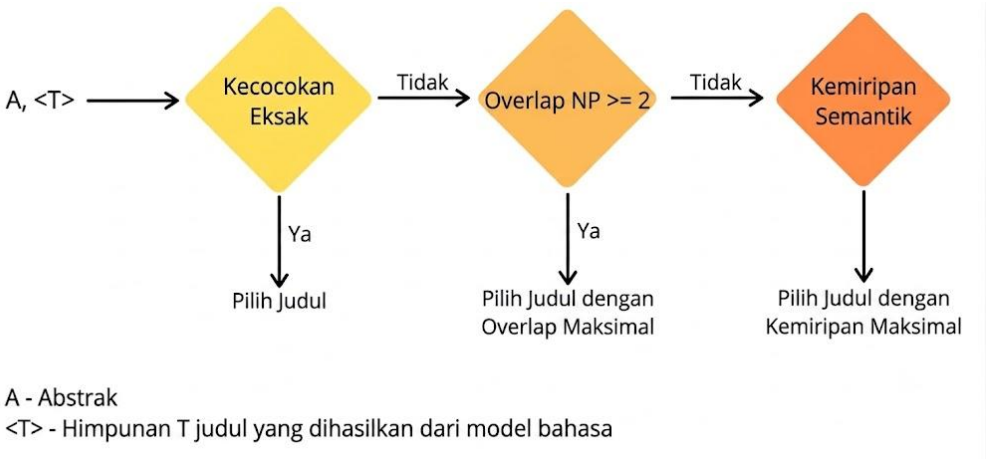
Komponen pertama dari modul Seleksi yang disebut Pencocokan Tepat memeriksa apakah judul yang dihasilkan cocok persis dengan kalimat yang ada dalam abstrak. Alasan kami untuk memeriksa kecocokan persis adalah karena judul tersebut akan benar secara tata bahasa, lancar, dan juga akan menggambarkan tema utama dokumen masukan. Namun, pencocokan persis judul dengan abstrak hanya membentuk persentase yang sangat kecil dari judul yang dihasilkan, terutama di domain ilmiah seperti domain kami. Dalam kasus korpus arXiv kami, hanya 4% dari abstrak dataset pelatihan yang memiliki judul yang teksnya sudah ada dalam abstrak. Meskipun demikian, pendekatan kami pertama-tama memeriksanya karena keuntungan yang disebutkan di atas.

Komponen kedua dari modul Seleksi didasarkan pada tumpang tindih NP antara judul dan abstrak yang mendorong prosedur seleksi kami ke arah sifat ekstraktif. Aturan ini membantu kami dalam memilih judul yang secara sintaksis lebih dekat dengan dokumen masukan daripada memilih judul yang sama sekali berbeda. Kami menerapkan pencarian Grid untuk menyesuaikan parameter ambang tumpang tindih NP.

Pada komponen terakhir dari modul Seleksi, kami memvektorisasi judul dan dokumen masukan menggunakan model yang telah dilatih sebelumnya. Pentransformasi Kalimat dan kemudian mengukur kedekatan semantik di antara mereka menggunakan kesamaan kosinus. Judul dengan kesamaan tertinggi dipilih. Algoritma 1 menguraikan pseudokode untuk komponen dalam alur kerja modul Seleksi berbasis aturan seperti yang dibahas di atas.

Tabel 14.1 Skor validasi untuk memilih komponen modul Seleksi.

Kriteria Seleksi	**R-1**	**R-2**	**R-L**
Acak (*Random*)	0.373	0.180	0.329
Pencocokan Eksak + Acak	0.375	0.186	0.332
Pencocokan Eksak + Frasa Nomina + Acak	0.380	0.187	0.341
Pencocokan Eksak + Frasa Nomina + Kemiripan Semantik	**0.388**	**0.190**	**0.342**



Gambar 14.3 Alur seleksi yang menunjukkan pemilihan judul dari kumpulan judul kandidat.

Modul Penyempurnaan

Dalam modul Penyempurnaan, kami melakukan fine-tuning pada model bahasa GPT-2 berukuran sedang yang dilatih secara terpisah dan urutannya diformat mirip dengan modul Generasi, dengan kata pemisah berupa ⟨untuk menghilangkan noise⟩. Untuk menyempurnakan model ini, kami membuat dataset dengan mempertimbangkan judul sampel dari output modul Generasi sebagai input ke modul Penyempurnaan, dan judul asli sebagai output yang diharapkan. Model ini seharusnya mempelajari pemetaan dari judul input ke judul asli dan mendorong kemampuan generasi menuju kebenaran mutlak. Pelatihan komponen ini melibatkan optimasi pada tujuan yang serupa dengan modul Generasi. Namun, tidak seperti pada modul Generasi, di sini kami menggunakan strategi decoding Greedy untuk

mengambil sampel kata pada setiap langkah waktu. Ini karena kami menginginkan output deterministik dari modul Penyempurnaan.

Algoritma 1. Pseudocode Modul Seleksi

```
syn, sem ← emptydictionary
for i = 1, 2, ..., T do
    if title_i dalam abstrak then
        return title_i
    end if
    syn[title_i] ← NP(title_i, abstrak)
    v_t, v_a = Vectorize(title_i, abstrak)
    sem[title_i] ← Cosine(v_t, v_a)
end for
if max(syn) ≥ 2 then
    title_k = max(syn)
    return title_k
end if
title_k = max(sem)
return title_k
```

Fungsi Penilaian

Fungsi Penilaian adalah langkah terakhir dalam alur kerja yang menghitung skor relevansi untuk judul dari modul Seleksi dan modul Penyempurnaan. Ini dihitung menggunakan kesamaan Jaccard pada himpunan NP antara judul dan abstrak input. Nilai penghalusan 0.0001 juga ditambahkan untuk menghindari kasus kesamaan Jaccard nol. Alasan kami menambahkan nilai penghalus ini ke pembilang adalah karena jika terdapat tumpang tindih NP nol antara judul terpilih/yang disempurnakan dan abstrak, maka alih-alih memberikan skor nol kepada kedua judul tersebut, kami tetap ingin memberikan penalti pada judul yang memiliki generasi NP positif palsu yang lebih tinggi antara judul terpilih dan yang disempurnakan. Berikut adalah persamaan untuk skor relevansi:

$$\text{relevancy score}(t) = \frac{\text{NP}(t) \cap \text{NP}(A) + 0.0001}{\text{NP}(t) \cup \text{NP}(A)} \tag{14.3}$$

di mana t, NP, dan A masing-masing mengacu pada judul, NP, dan abstrak.

Alasan kami menggunakan metrik tumpang tindih NP lagi pada langkah terakhir untuk menghitung skor relevansi adalah karena modul Penyempurnaan dapat menambahkan beberapa NP yang relevan atau dapat menghapus NP yang berlebihan dari judul. Kita dapat melihat ini melalui contoh acak yang ditampilkan pada Tabel 14.2 di mana output untuk setiap modul ditampilkan. Tabel tersebut menunjukkan abstrak, judul aktual, lima judul dari modul Generasi, judul terpilih dari output Seleksi (ditampilkan dalam warna hijau) dan judul yang disempurnakan untuk judul terpilih yang diproses oleh modul Penyempurnaan. Tumpang tindih NP ditandai dengan warna merah. Kita dapat mengamati bahwa NP "Person Re-

Identification" ditambahkan ke judul setelah langkah Penyempurnaan. Ini menggantikan kata "Re-Id" yang tidak masuk akal secara semantik, dengan NP yang benar yaitu "Re-Identification". Modul Penyempurnaan mampu mengidentifikasi kata pengganti dengan benar. Skor relevansi pun meningkat untuk judul ini, sehingga terpilih dalam alur kerja otomatis. Ini menegaskan kembali pentingnya tumpang tindih NP pada fase terakhir pemilihan judul akhir.

Tabel 14.2 Hasil dari modul Seleksi dan Penyempurnaan.

Abstrak: Mempelajari metrik jarak antara pasangan contoh sangat penting untuk pengenalan visual, terutama untuk identifikasi ulang orang (Re-Id). Baru-baru ini, loss kontrastif dan triplet diusulkan untuk meningkatkan daya diskriminatif fitur yang dipelajari secara mendalam, dan telah mencapai keberhasilan yang luar biasa. Seperti yang dapat dilihat, loss kontrastif atau triplet hanyalah satu kasus khusus dari hubungan jarak Euclidean di antara sampel pelatihan ini. Oleh karena itu, kami mengusulkan algoritma penyematan Laplacian grafik terstruktur, yang dapat merumuskan semua hubungan jarak terstruktur ini ke dalam bentuk Laplacian grafik. Metode yang diusulkan dapat memanfaatkan sepenuhnya hubungan jarak terstruktur di antara sampel pelatihan ini, dengan grafik lengkap yang dibangun. Selain itu, formulasi ini membuat metode kami mudah diimplementasikan dan sangat efektif. Ketika menyematkan algoritma yang diusulkan dengan loss softmax untuk pelatihan CNN, metode kami dapat memperoleh fitur mendalam yang jauh lebih kuat dan diskriminatif dengan dispersi antar-pribadi dan kekompakan intra-pribadi, yang sangat penting untuk Re-Id orang. Kami mengilustrasikan efektivitas metode yang kami usulkan di atas tiga jaringan populer, yaitu AlexNet, DGDNet, dan ResNet50, pada empat dataset benchmark Re-Id yang banyak digunakan baru-baru ini. Metode yang kami usulkan mencapai kinerja terbaik.
Aktual: Pembelajaran Fitur Mendalam melalui Penyematan Laplacian Grafik Terstruktur untuk Re-Id Orang
Judul-1: Penyematan Laplacian Grafik Terstruktur untuk Re-Id Orang Mendalam/**RS**: 0.03 **Judul-2:** Penyematan Laplacian Grafik Terstruktur dari Kerugian Kontrastif dan Triplet **Judul-3:** Algoritma Grafik Terstruktur untuk Re-Id Orang Mendalam
Judul-4: Relasi Jarak Terstruktur di antara Contoh Pelatihan
Judul-5: Penyematan Laplacian Grafik Terstruktur untuk Re-Id Orang Mendalam
Judul yang Disempurnakan: Penyematan Laplacian Grafik Terstruktur untuk Re-Id Orang/**RS**: 0.07

Modul Peringkas

Modul peringkas didasarkan pada penyempurnaan model T5 yang telah dilatih sebelumnya. Model T5 yang telah dilatih sebelumnya dilatih pada korpus yang sangat besar yang disebut *Colossal Clean Crawled Corpus* (C4) dan disempurnakan pada tugas peringkasan teks (di antara berbagai tugas NLP yang disempurnakan), sehingga cocok untuk tugas kita. Hal ini karena model tersebut sudah memiliki parameter yang dipelajari untuk meringkas informasi tertentu dan model yang telah disempurnakan hanya perlu mempelajari kosakata domain kita dan spesifikasi tugas peringkasan kita seperti panjang, gaya, dll.

Kami menggunakan dataset artikel pembelajaran untuk menyempurnakan modul peringkas kami. Kami menggunakan awalan yang sama, yaitu *summarize*:, yang digunakan sebelumnya pada model T5 yang telah dilatih sebelumnya. Masukan ke model adalah bagian

Pendahuluan dari artikel pembelajaran yang diawali dengan kata kunci *summarize*: dan keluarannya adalah abstrak dari artikel yang sama.

Secara formal, pertimbangkan $D_{\text{abstract}}(I, A)$ dengan ``$N$" titik data sedemikian sehingga $D_{\text{abstract}} = (I_1, A_1), (I_2, A_2), \ldots, (I_N, A_N)$ dan, $I$ dan $A$ masing-masing mewakili himpunan Pendahuluan dan Abstrak. Kita menambahkan awalan pada setiap titik data dengan teks ``*ringkas:*" sehingga titik data menjadi $I = \text{summarize: } i_1, i_2, \ldots, i_{l_1}\langle/s\rangle, A = o_1, o_2, \ldots, o_{l_2}\langle/s\rangle$, $A = o_1, o_2, \ldots, o_{l_2}\langle/s\rangle$, di mana $i, o, \langle/s\rangle, l_1$ dan l_2 masing-masing adalah token masukan, token keluaran, token akhir string, panjang Pendahuluan dan Abstrak. Tujuan pelatihan di sini adalah untuk mempelajari pemetaan dari $I \rightarrow A$ yang merupakan probabilitas bersyarat $P(A|I)$. Ini dilakukan dengan memaksimalkan tujuan seperti yang didefinisikan dalam persamaan berikut:

$$P(D_{\text{abstract}}) = \prod_{i=1}^{N} \prod_{j=1}^{l_2} P\left(o_j \middle| I, o_1, \ldots, o_{j-1}; \theta\right). \tag{14.4}$$

Di sini, θ adalah parameter model.

Model Summarizer dioptimalkan pada kerugian cross-entropy seperti yang didefinisikan dalam Persamaan (14.2). Untuk decoder, kami menggunakan strategi pengambilan sampel Top-k, Top-p. Pada inferensi, ringkasan dihasilkan berdasarkan input I sedemikian rupa sehingga output oj dihasilkan pada setiap langkah waktu hingga akhir token string $\langle/s\rangle$ atau panjang output maksimum tercapai.

14.4 EKSPERIMEN DAN HASIL UNTUK PEMBUATAN JUDUL

Pada bagian ini, kami menjelaskan dataset, detail implementasi model Pembuatan Judul kami, algoritma dasar, dan hasilnya.

Dataset

Kami melatih model Generasi Judul pada dataset arXiv.a Dataset arXiv terdiri dari objek JSON dari 41 ribu artikel pembelajaran dari domain seperti ML, CL, NER, AI dan CV dari tahun 1980 hingga 2018. Kami hanya mempertimbangkan abstrak dan judul dari dataset yang membentuk pasangan input dan output untuk model generatif kami.

Tabel 14.3 Statistik dataset untuk model Pembuatan Judul.

Dataset	Dokumen	Rata-rata Panjang Abstrak	Rata-rata Panjang Judul	Persentase Kompresi
arXiv	41 k	150.58	8.71	6.71
ACL	10.8 k	116.08	9.38	9.22
ICMLA	448	155.26	10.04	7.06

Kami membagi data kami menjadi set pelatihan (75%), set pengujian (25%) dan set validasi (5% sampel dari set pelatihan). Kami juga memvalidasi model kami pada dataset lain yang belum pernah dilihat sebelumnya seperti data *Association for Computational Linguistics* (ACL) dan data *International Conference on Machine Learning and Applications* (ICMLAs).

Tabel 14.4 Parameter model.

Parameter	Nilai
Ukuran *batch*	32
Epoch	4
Fungsi *loss*	*Cross entropy*
Learning rate	2e–05
Jadwal *learning rate*	*Cosine* dengan *restart*

Tabel 14.3 mencantumkan beberapa properti statistik penting untuk ketiga dataset tersebut. Panjangnya berada pada tingkat kata. Persentase kompresi menunjukkan rasio panjang judul terhadap panjang ringkasan.

Detail Implementasi

Untuk menyempurnakan model GPT-2 berukuran sedang sebagai bagian dari modul Generasi dan modul Penyempurnaan kami, kami menggunakan pustaka HuggingFace Transformers[b] di PyTorch.[c]. Semua hyper-parameter yang diperlukan yang disebutkan dalam Tabel 14.4 dipilih berdasarkan parameter berkinerja terbaik pada set validasi. Selain itu, kami bereksperimen dengan berbagai nilai k dan p untuk pengambilan sampel Top-k, Top-p kami dan menemukan $k = 2$ dan $p = 0.8$ memberikan skor terbaik pada pembagian validasi. Model dilatih pada instance gratis Google Colab[d] dan membutuhkan waktu sekitar 10 jam pada satu server. Tabel 14.5 mengacu pada skor eksperimental progresif dari beberapa model eksperimental yang kami coba sebelum menetapkan model akhir.

Tabel 14.5 Skor ROUGE dan BLEU untuk ketiga dataset pada eksperimen inkremental metode kami.

	arXiv						
	ROUGE			**BLEU**			
Metode Kami	1	2	L	1	2	3	4
*Gen	0.336	0.154	0.306	0.324	0.219	0.179	0.102
*Gen + PP	0.368	0.175	0.321	0.348	0.238	0.188	0.109
*Gen + PP + Sel	**0.376**	**0.188**	**0.336**	**0.358**	**0.250**	**0.209**	**0.124**
	ACL						
*Gen	0.296	0.113	0.268	0.280	0.173	0.136	0.072
*Gen + PP	0.328	0.134	0.295	0.317	0.201	0.162	0.090
*Gen + PP + Sel	**0.340**	**0.156**	**0.308**	**0.321**	**0.219**	**0.187**	**0.108**
	ICMLA						
*Gen	0.361	0.159	0.319	0.315	0.209	0.169	0.101
*Gen + PP	0.384	0.181	0.346	0.334	0.227	0.176	0.097
*Gen + PP + Sel	**0.404**	**0.189**	**0.356**	**0.395**	**0.271**	**0.216**	**0.123**

Semua skor yang ditandai dengan * adalah skor rata-rata dari lima kali pengujian pada bagian pengujian.

Garis Dasar

Kami membandingkan hasil kami dengan beberapa algoritma sebagai berikut:

PREFIX. Dalam metode ini, kami memilih kalimat pertama dalam abstrak sebagai judul.

TextRank. Dalam algoritma ini, Mihalcea dan Tarau mengusulkan representasi berbasis grafik pada teks masukan, di mana kalimat dimodelkan sebagai simpul dalam grafik dengan skema pembobotan yang ditentukan. Algoritma PageRank kemudian diterapkan untuk memberi skor pada setiap simpul dalam grafik dan kalimat dengan skor tertinggi dipilih sebagai judul abstrak yang diberikan. Kami menggunakan paket Python Gensime untuk tujuan implementasi.

LexRank. Dalam algoritma ini, Erkan dan Radev mengusulkan pendekatan tanpa pengawasan untuk peringkasan teks berdasarkan penilaian sentralitas berbasis grafik dari kalimat. Kami memilih kalimat dengan skor tertinggi sebagai judul abstrak yang diberikan. Kami menggunakan paket Python Sumyf untuk tujuan implementasi.

Bi-GRU dengan Attention. Kami menggunakan arsitektur encoder-decoder berbasis GRU (Jaringan Sequence-to-Sequence) dengan attention untuk menghasilkan judul untuk abstrak yang diberikan. Kami menggunakan parameter/rasio teacher forcing untuk memutuskan keseimbangan antara memasukkan ground truth dibandingkan dengan memperlakukan output sebelumnya sebagai input untuk langkah saat ini. Selama pelatihan, kami memasukkan kata-kata yang dihasilkan alih-alih kata-kata sebenarnya, kami melakukan ini secara khusus 30% dari waktu. Semua parameter dipilih berdasarkan parameter berkinerja terbaik pada set validasi.

Evaluasi zero-shot dengan GPT-2. Kami menggunakan model bahasa pra-terlatih GPT-2 berukuran sedang dalam kasus ini. Untuk menginduksi perilaku peringkasan, kami menambahkan kata kunci TL; DR: di akhir urutan input.

Evaluasi dan Hasil

Kami mengevaluasi pendekatan kami dengan membandingkannya dengan algoritma dasar seperti yang dijelaskan di atas. Kami menggunakan metrik ROUGE dan BLEU untuk perbandingan karena ini adalah metrik evaluasi standar saat ini dalam sistem peringkasan. Metrik ini pada dasarnya bergantung pada tumpang tindih n-gram antara teks yang dihasilkan dan teks sebenarnya dengan ROUGE mengukur recall dan BLEU mengukur presisi.

Tabel 14.6 menunjukkan hasil evaluasi otomatis kami pada dataset arXiv test split, ACL, dan ICMLA. Kita dapat melihat bahwa pendekatan kami memiliki skor ROUGE dan BLEU yang tinggi di ketiga dataset tersebut dibandingkan dengan algoritma dasar. Karena pendekatan kami menghasilkan sepasang judul dengan skor relevansi, kami juga melakukan evaluasi manusia untuk membantu memvalidasi seberapa baik seleksi manusia berkorelasi dengan penerapan pipeline otomatis sepenuhnya. Kami tidak membuat evaluasi manusia terpisah untuk memvalidasi kualitas judul karena pengguna dapat memilih salah satu/tidak satupun opsi jika dia tidak setuju dengan salah satu opsi judul yang diberikan. Untuk evaluasi, kami memiliki 22 evaluator manusia yang terdiri dari lulusan S1, S2, dan mahasiswa S3 yang mahir dalam bidang teknis yang mewakili dataset kami. Para evaluator diberikan sampel 25 abstrak

beserta pilihan judul dari modul Seleksi dan Penyempurnaan serta pilihan "salah satu/tidak satupun".

Tabel 14.6 Skor ROUGE dan BLEU untuk ketiga dataset.

	arXiv						
	ROUGE			**BLEU**			
Metode	**1**	**2**	**L**	**1**	**2**	**3**	**4**
PREFIX	0.228	0.099	0.206	0.160	0.103	0.091	0.047
TextRank	0.137	0.039	0.130	0.083	0.043	0.036	0.015
LexRank	0.204	0.08	0.186	0.131	0.079	0.067	0.033
*Bi-GRU w/attention	0.211	0.045	0.162	0.250	0.189	0.163	0.132
*Zero-shot GPT-2	0.123	0.044	0.170	0.058	0.033	0.030	0.013
*Metode Kami	**0.376**	**0.188**	**0.336**	**0.358**	**0.250**	**0.209**	**0.124**
			ACL				
PREFIX	0.234	0.099	0.213	0.154	0.097	0.086	0.044
TextRank	0.142	0.038	0.133	0.087	0.044	0.037	0.015
LexRank	0.221	0.089	0.201	0.143	0.088	0.077	0.0396
*Bi-GRU w/attention	0.230	0.56	0.183	0.235	0.183	0.161	0.130
*Zero-shot GPT-2	0.120	0.035	0.154	0.061	0.032	0.030	0.014
*Metode Kami	**0.340**	**0.156**	**0.308**	**0.321**	**0.219**	**0.187**	**0.108**
			ICMLA				
PREFIX	0.211	0.081	0.185	0.150	0.090	0.075	0.036
TextRank	0.151	0.042	0.139	0.094	0.048	0.038	0.014
LexRank	0.202	0.074	0.179	0.136	0.079	0.067	0.032
*Bi-GRU w/attention	0.198	0.075	0.161	0.179	0.181	0.158	0.126
*Zero-shot GPT-2	0.115	0.029	0.145	0.063	0.034	0.028	0.013
*Metode Kami	**0.404**	**0.189**	**0.356**	**0.395**	**0.271**	**0.216**	**0.123**

Hasil evaluasi ini disajikan sebagai Matriks Kebingungan pada Tabel 14.7. Kita dapat melihat bahwa presisi dan recall masing-masing adalah 0.76 dan 0.86 untuk Seleksi dan 0.75 dan 0.60 untuk Penyempurnaan. Tabel 14.8 menunjukkan beberapa contoh pendekatan pembuatan judul kami yang diambil dari kumpulan yang disajikan untuk evaluasi manusia. Contohnya termasuk judul abstrak, aktual, terpilih, dan disempurnakan beserta skor relevansinya.

Dalam semua contoh ini, penilaian manusia dalam hal memilih judul akhir sesuai dengan pilihan akhir kami berdasarkan skor relevansi. Pengguna tidak diberikan judul aktual untuk menghindari bias saat memilih judul yang disukai. Kami juga membuat pengamatan berikut untuk judul yang dihasilkan oleh pendekatan kami:

- Model mereproduksi kata majemuk yang dipisahkan tanda baca dengan benar hampir sepanjang waktu. Lihat contoh 1 pada Tabel 14.8.
- Pipeline kami mempertahankan struktur frasa spesifik korpus saat menghasilkan judul. Pada contoh 2 di Tabel 14.8, kita amati bahwa modul Refinement mampu

merekonstruksi frasa "*Transfer Learning*" dengan benar dari frasa parsial judul yang dipilih.

Tabel 14.7 Matriks kebingungan untuk evaluasi manusia.

	Manusia		
Model	**Seleksi**	**Penyempurnaan**	**Keduanya/Tidak satu pun**
Seleksi	13	2	0
Penyempurnaan	4	6	0

14.5 EKSPERIMEN DAN HASIL UNTUK MODUL PERINGKAS

Pada bagian ini, kami menjelaskan dataset, detail implementasi, algoritma dasar, dan hasil dari modul peringkas kami.

Dataset

Kami melatih model kami pada subset data acak yang diambil dari dataset.[g] $arXiv$. Dataset ini terdiri dari objek JSON sekitar 42 ribu artikel pembelajaran dengan semua informasi yang diuraikan di bawah bagian yang relevan. Kami mempertimbangkan bagian Pendahuluan dan Abstrak dari dataset sebagai pilihan pasangan input/output kami. Kami membagi data kami menjadi set pelatihan sebanyak 30 ribu sampel, set pengujian sebanyak 6440 sampel, dan set validasi sebanyak 6436 sampel.

Tabel 14.8 Contoh keluaran dari modul Seleksi dan Pemurnian untuk abstrak masukan yang diberikan.

Abstrak: Artikel ini menyajikan pendekatan baru untuk klasifikasi sentimen multibahasa dalam teks pendek. Ini adalah tugas yang menantang karena jumlah data pelatihan dalam bahasa selain Inggris sangat terbatas. Pendekatan multibahasa yang diusulkan sebelumnya biasanya memerlukan penetapan korespondensi ke bahasa Inggris yang mana pengklasifikasi yang kuat sudah tersedia. Sebaliknya, metode kami tidak memerlukan pengawasan seperti itu. Kami memanfaatkan sejumlah besar data yang diawasi secara lemah dalam berbagai bahasa untuk melatih jaringan konvolusional multi-lapisan dan menunjukkan pentingnya menggunakan pra-pelatihan jaringan tersebut. Kami mengevaluasi pendekatan kami secara menyeluruh pada berbagai dataset multibahasa, termasuk benchmark prediksi sentimen SemEval-2016 terbaru (Tugas 4), di mana kami mencapai kinerja terbaik. Kami juga membandingkan kinerja model kami yang dilatih secara individual untuk setiap bahasa dengan varian yang dilatih untuk semua bahasa sekaligus. Kami menunjukkan bahwa model yang terakhir mencapai kinerja yang sedikit lebih buruk — tetapi masih dapat diterima — jika dibandingkan dengan model bahasa tunggal, sambil mendapatkan manfaat dari sifat generalisasi yang lebih baik di berbagai bahasa.

Judul Asli: Memanfaatkan Sejumlah Besar Data yang Diawasi Secara Lemah untuk Klasifikasi Sentimen Multibahasa

Judul Terpilih: Klasifikasi Sentimen Multibahasa dengan Data yang Diawasi Secara Lemah dalam Teks Pendek/**RS: 0.11**

Judul yang Disempurnakan: Klasifikasi Sentimen Multibahasa dengan Data yang Diawasi Secara Lemah/**RS: 0.05**

Abstrak: Artikel ini mengusulkan kerangka kerja transfer online untuk menangkap interaksi antar agen dan menunjukkan bahwa pembelajaran transfer saat ini dalam pembelajaran penguatan adalah kasus khusus dari transfer online. Lebih lanjut, artikel ini mengkarakterisasi ulang metode agen-mengajar-agen yang ada sebagai transfer online dan memeriksa salah satu metode pengajaran tersebut dalam tiga cara. Pertama, konvergensi Q-learning dan Sarsa dengan representasi tabular dengan anggaran terbatas dibuktikan. Kedua, konvergensi Q-learning dan Sarsa dengan aproksimasi fungsi linier ditetapkan. Ketiga, kami menunjukkan bahwa kinerja asimtotik tidak dapat dirugikan melalui pengajaran. Selain itu, semua hasil teoritis divalidasi secara empiris.

Judul Aktual: Pembelajaran Transfer Online dalam Domain Pembelajaran Penguatan

Judul Terpilih: Transfer Online dalam Pembelajaran Penguatan: Pendekatan Baru/**RS: 0.09**

Judul yang Disempurnakan: Pembelajaran Transfer untuk Pembelajaran Penguatan/**RS: 0.10**

Tabel 14.9 Skor ROUGE antara teks masukan dan abstrak.

Teks Masukan	R-1	R-2	R-L
Pendahuluan	**0.260**	**0.112**	**0.276**
Pendahuluan + Kesimpulan	0.209	0.102	0.256

Tabel 14.10 Statistik data untuk modul Summarizer.

Pembagian Data	Sampel	Rata-rata Pendahuluan	Rata-rata Abstrak
Latih	30.000	889	295
Validasi	6.436	860	172
Uji	6.440	845	174

Sebelum sampai pada kesimpulan memilih Pendahuluan sebagai input untuk penyempurnaan model T5, kami juga bereksperimen dengan Pendahuluan + Kesimpulan sebagai proksi untuk menangkap esensi utama artikel. Kami menggunakan metrik ROUGE untuk membandingkan perbedaan output yang diberikan berbagai jenis teks input. Tabel 14.9 menunjukkan skor eksperimen yang relevan untuk hal yang sama.

Kita dapat melihat skor ROUGE yang lebih tinggi ketika teks input adalah Pendahuluan dibandingkan dengan Pendahuluan + Kesimpulan. Hipotesis kami untuk perilaku ini adalah bahwa Kesimpulan biasanya berisi informasi tambahan seperti arah kerja di masa depan, pengamatan eksperimen, dll., yang biasanya tidak mewakili ringkasan artikel. Setelah menyelesaikan Pendahuluan dan Abstrak sebagai pasangan input dan output kami dan kemudian menyempurnakan model T5 kami. Tabel 14.10 mencantumkan beberapa properti statistik penting untuk dataset. Panjangnya berada pada tingkat kata.

Detail Implementasi

Kami menggunakan pustaka HuggingFace Transformers di PyTorch untuk menyempurnakan model dasar T5 kami. Jumlah blok Transformer di encoder dan decoder adalah 12, jumlah head self-attention adalah 12, dan jumlah total parameter model pra-terlatih adalah 220 M (220 Juta). Semua hyper-parameter model yang digunakan selama proses fine-tuning disebutkan dalam Tabel 14.11. Kami juga bereksperimen dengan berbagai nilai k dan p untuk strategi sampling Top-k, Top-p kami dan menemukan bahwa $k = 50$ dan

$p = 0.85$ memberikan skor terbaik pada set validasi kami. Model dilatih pada instance gratis Google Colab.

Tabel 14.11 Parameter hiper modul peringkas.

Parameter	Nilai
Ukuran *batch*	1
Epoch	2
Loss	*Cross entropy*
Learning rate	3e–4
Langkah akumulasi gradien	16

Evaluasi dan Hasil

Kami membandingkan model yang telah disempurnakan dengan beberapa algoritma dasar seperti yang dibahas berikut ini:

Zero-shot GPT-2. Dalam hal ini, kami menggunakan model bahasa GPT-2 pra-terlatih berukuran sedang. Untuk menginduksi perilaku peringkasan, kami menambahkan kata kunci TL;DR: di akhir urutan input.

GPT-2 yang disempurnakan. Kami menyempurnakan model bahasa GPT-2 berukuran sedang pada dataset arXiv kami yang terdiri dari Introduction dan Abstract sebagai pasangan input dan output kami (dataset yang sama yang digunakan untuk menyempurnakan modul Summarizer berbasis T5 kami).

Zero-shot T5. Dalam hal ini, kami menggunakan model dasar T5 pra-terlatih. Untuk menginduksi perilaku peringkasan, kami menambahkan kata kunci *summarize*: di awal urutan input.

Ukuran input default dari model GPT-2 yang telah dilatih sebelumnya dan model T5 yang telah dilatih sebelumnya adalah 512 token. Oleh karena itu, kami menggunakan panjang input yang sama untuk melakukan fine-tuning pada model GPT-2 dan T5 agar baseline dapat dibandingkan. Kami menggunakan panjang output 256 token untuk semua baseline di atas dan modul Summarizer T5 yang telah kami fine-tuning. Kami menggunakan ROUGE sebagai pilihan metrik evaluasi kami yang membandingkan abstrak yang dihasilkan dengan abstrak ground truth. Tabel 14.12 menunjukkan hasilnya. Kita melihat bahwa, dalam pengaturan yang diberikan, model T5 yang telah kami fine-tuning berkinerja jauh lebih baik daripada semua metode lain yang disebutkan. Bukti empiris ini semakin memperkuat keyakinan kami bahwa T5 berkinerja lebih baik pada tugas kami karena model tersebut telah mempelajari tugas peringkasan selama pra-pelatihan. Oleh karena itu, kami memilih model T5 yang telah di-fine-tuning untuk modul Summarizer kami.

Tabel 14.12 Skor ROUGE untuk perangkum.

Metode	R-1	R-2	R-L
GPT-2 *Zero-shot*	0.214	0.035	0.168
GPT-2 *Fine-tuned*	0.289	0.067	0.238
T5 *Zero-shot*	0.231	0.053	0.205

T5 *Fine-tuned*	**0.327**	**0.085**	**0.255**

Tabel 14.13 Skor ROUGE untuk varian panjang input/output Summarizer.

Metode	**R-1**	**R-2**	**R-L**
T5 *Fine-tuned* w/ 512/256	0.327	0.085	0.255
T5 *Fine-tuned* w/ 700/256	0.332	0.099	0.273
T5 *Fine-tuned* w/ 900/256	**0.366**	**0.110**	**0.294**

Kami selanjutnya bereksperimen dengan parameter panjang input dari model T5. Model T5 yang telah dilatih sebelumnya dilatih dengan panjang input maksimum 512 token, namun untuk kasus penggunaan kami, panjang input rata-rata sekitar 850 kata. Oleh karena itu, kami bereksperimen dengan menimpa parameter panjang maksimum dan melatih model T5 untuk panjang input yang semakin tinggi.

Tabel 14.13 menunjukkan skor evaluasi untuk modul Summarizer pada set pengujian dengan panjang input yang bervariasi yaitu 512, 700, dan 900 token untuk ketiga model tersebut. Kita melihat bahwa skor ROUGE meningkat seiring dengan peningkatan panjang input. Hipotesis kami untuk perilaku ini adalah bahwa model sekarang memiliki lebih banyak informasi untuk diproses dan menghasilkan ringkasan yang relevan. Selain itu, dengan mengamati Pendahuluan secara saksama, kita melihat bahwa penulis membahas kontribusi utama artikel sebagian besar di segmen teks akhir Pendahuluan, sehingga meningkatkan akurasi peringkasan seiring dengan panjangnya. Mempertimbangkan semua hal di atas, kami memilih varian input 900 token dari model T5 yang telah disempurnakan sebagai pilihan kami untuk Summarizer.

14.6 PEMBUATAN JUDUL HIERARKIS UNTUK JALUR PEMBELAJARAN

Pada bagian ini, kami menjelaskan dataset, eksperimen, dan hasil untuk eksperimen jalur pembelajaran. Kami mengusulkan pendekatan pembuatan judul hierarkis untuk jalur pembelajaran. Setelah kami memiliki judul dan ringkasan untuk setiap sumber daya dalam suatu jalur pembelajaran, kami menerapkan ensemble dari empat pendekatan ekstraktif untuk menghasilkan k kata kunci teratas yang mewakili jalur pembelajaran. Kemudian kami menggunakan daftar kata kunci yang diberi peringkat dari semua pendekatan dan menggabungkan output ini menggunakan fungsi irisan. Fungsi irisan mengembalikan daftar kata kunci yang diberi peringkat yang ditemukan di keempat metode tersebut. Kami memperlakukan pencetak skor teratas 1 sebagai judul representatif untuk seluruh jalur pembelajaran.

Dataset

Karena model kami dilatih pada korpus arXiv, kami ingin memilih jalur pembelajaran dari domain teknis yang serupa untuk eksperimen kami. Oleh karena itu, kami mempertimbangkan empat books^{h} teks Pembelajaran Mesin sebagai pilihan dataset kami. Kami membuat sampel empat bab berurutan secara rata-rata di semua buku. Ini analog dengan jalur pembelajaran dalam pembelajaran daring karena bab-bab dalam buku teks

sebagian besar disusun sedemikian rupa sehingga koheren, berbagi topik global yang sama, dan juga bersifat progresif. Kami mendapatkan total 54 bab yang mencakup 13 jalur pembelajaran.

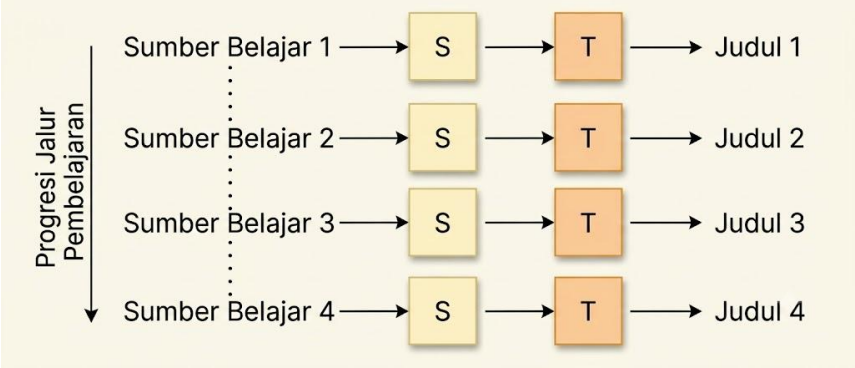
Eksperimen

Kami melakukan dua set eksperimen — Pembuatan judul tingkat sumber daya dan Pembuatan judul tingkat jalur pembelajaran.

Pembuatan Judul Tingkat Sumber Daya

Untuk setiap sumber daya dalam jalur pembelajaran, kami memanggil modul Penyusun Ringkasan untuk mendapatkan ringkasan. Kemudian kami meneruskan ringkasan yang dihasilkan ini ke model Pembuatan Judul untuk mendapatkan judul untuk sumber daya yang diberikan. Gambar 14.4 menunjukkan representasi visual dari eksperimen ini. Di sini, S dan T mewakili modul Penyusun Ringkasan dan model Pembuatan Judul.

Selain itu, setelah melakukan beberapa pemeriksaan statistik pada judul yang dihasilkan, kami menemukan bahwa judul yang dihasilkan oleh model kami rata-rata sekitar sembilan kata, sedangkan panjang rata-rata kebenaran sebenarnya (judul bab yang sebenarnya) sekitar dua kata. Alasan model kami menghasilkan judul deskriptif yang panjang adalah karena model kami dilatih pada judul artikel pembelajaran yang cenderung sedikit deskriptif dan panjang dibandingkan dengan judul bab buku yang lebih berorientasi pada kata kunci. Judul deskriptif ini sangat cocok untuk aplikasi kami dalam menghasilkan pratinjau jalur pembelajaran. Namun, pergeseran distribusi panjang judul membuat kami mempertimbangkan kembali metrik evaluasi dan menggunakan ukuran perbandingan semantik daripada skor ROUGE yang didasarkan pada tumpang tindih kata sebenarnya. Kami menggunakan kesamaan semantik antara judul yang dihasilkan/aktual dan teks bab aktual sebagai metrik evaluasi kami. Metrik ini juga merupakan ukuran standar untuk membandingkan teks.



Gambar 14.4 Alur pembuatan judul tingkat sumber daya.

Untuk menghitung kesamaan semantik, pertama-tama kami memvektorisasi sepotong teks yang diberikan menggunakan Sentence Transformer yang telah dilatih sebelumnya dan kemudian meneruskan vektor yang relevan ke fungsi kesamaan kosinus untuk mendapatkan skor kesamaan atau kedekatan semantik.

Rata-rata kesamaan semantik antara judul bab aktual dan isi bab adalah 37,27%, sedangkan rata-rata skor kesamaan semantik antara judul yang dihasilkan dan isi bab mencapai 44,99%. Ini menunjukkan bahwa judul yang kami hasilkan akurat dan relevan. Tabel 14 menunjukkan beberapa contoh untuk sampel yang dihasilkan dan aktual beserta skor kesamaan semantik antara judul dan isi bab yang ditulis dengan warna merah. Kita dapat melihat bahwa judul yang dihasilkan oleh model kami lebih representatif terhadap isi sumber daya pembelajaran. Kita juga dapat melihat peningkatan persentase yang signifikan yang diperoleh oleh pendekatan pembuatan judul kami. Rata-rata peningkatan persentase untuk semua judul yang dihasilkan oleh pendekatan kami adalah 7,72%.

Pembuatan Judul Tingkat Jalur Pembelajaran

Dalam percobaan ini, kami menghasilkan judul untuk jalur pembelajaran dalam beberapa kata. Kami membangun berdasarkan output yang dihasilkan dari eksperimen sebelumnya. Dengan menggunakan ringkasan dan judul yang dihasilkan untuk semua sumber belajar dalam suatu jalur pembelajaran, kami mengimplementasikan ensemble dari empat pendekatan ekstraktif untuk menghasilkan kata kunci top-k yang mewakili jalur tersebut. Tabel 14.15 menunjukkan metode yang kami gunakan sebagai bagian dari ensemble untuk mengekstrak kata kunci global untuk suatu jalur. Keempat metode tersebut merupakan kombinasi berbeda dari metode ekstraksi kata kunci dan teks masukan yang digunakan. Kami menggunakan YAKE dan TextRank untuk ekstraksi kata kunci atas abstrak dan judul yang dihasilkan untuk sumber belajar.

Selanjutnya, kami memberi peringkat kata kunci ini berdasarkan kedekatan semantiknya dengan sumber daya yang mendasarinya yang dipetakan ke jalur tersebut. Untuk itu, pertama-tama kami menyematkan kata kunci dan abstrak yang dihasilkan ke dalam representasi padat menggunakan Sentence Transformers dan kemudian menghitung kesamaan kosinus dari setiap kata kunci dengan abstrak yang digabungkan di semua sumber daya.

Tabel 14.14 Judul aktual dan yang dihasilkan untuk sumber daya pembelajaran.

Judul Aktual	Judul Generasi	Persen Keuntungan (%)
Pembelajaran efisien 39.80%	Algoritma pembelajaran dengan skala 58.89%	19.09
Prediktor linier 40.45%	Prediktor linier 40.45%	0
Metode kernel 53.32%	Pendekatan kernel untuk pemodelan komputasi 54.59%	1.27
Pengelompokan 31.04%	Pengelompokan dan pembagian kesamaan 51.35%	20.31
Mesin vektor pendukung 41.05%	Mesin vektor pendukung 43.47%	2.42

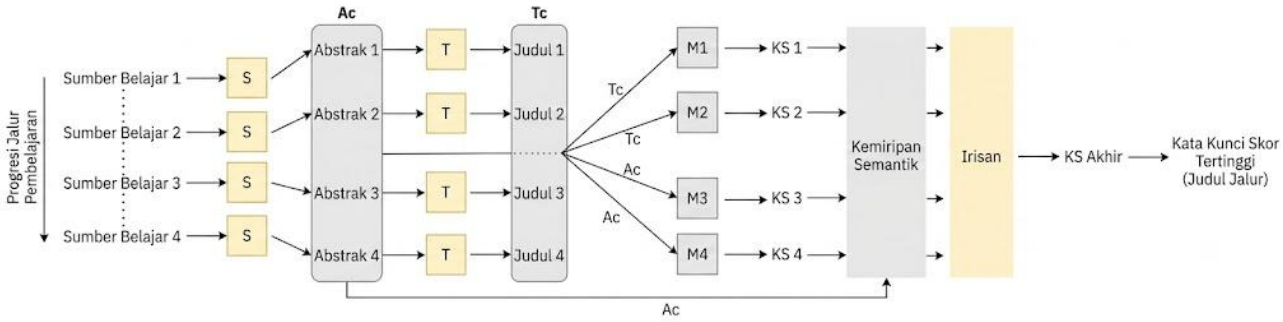
Tabel 14.15 Metode untuk mengekstrak judul jalur.

Metode	Teks Masukan	Teknik Ekstraksi Kata Kunci
1	Judul yang dihasilkan	YAKE
2	Judul yang dihasilkan	TextRank

3	Abstrak yang dihasilkan	YAKE
4	Abstrak yang dihasilkan	TextRank

Setelah kami mendapatkan daftar semua kata kunci yang diberi peringkat dari semua pendekatan untuk jalur tersebut, kami memilih serangkaian kata kunci kandidat yang umum untuk keempat metode tersebut. Dari kumpulan ini, kami memilih kata kunci dengan skor kesamaan semantik tertinggi sebagai pilihan judul representatif untuk seluruh jalur pembelajaran. Gambar 14.5 menunjukkan representasi visual dari eksperimen ini.

Tabel 14.16 menunjukkan daftar judul sumber daya aktual dan judul jalur pembelajaran yang dihasilkan oleh pendekatan kami untuk beberapa jalur yang diambil sampelnya. Kita dapat melihat bahwa judul yang dihasilkan oleh pendekatan yang kami usulkan untuk jalur pembelajaran dapat diinterpretasikan dan mencerminkan sumber daya yang dipetakan. Hal ini berlaku untuk semua jalur pembelajaran dalam dataset kami.



Gambar 14.5 Alur pembuatan judul tingkat jalur.

Catatan: S, T, M_i, KS, Ac dan Tc masing-masing adalah modul Summarizer, model Pembuatan Judul, metode ke-i untuk mengekstrak kata kunci, Kumpulan Kata Kunci, versi gabungan Abstrak dan Judul.

Tabel 14.16 Judul yang dihasilkan untuk jalur pembelajaran.

Jalur	Judul sumber daya aktual	Judul jalur yang dihasilkan
1	1.1. Jaringan maju dalam (*Deep forward network*)	Jaringan saraf feedforward
	1.2. Regularisasi untuk pembelajaran mendalam	
	1.3. Optimasi untuk melatih model mendalam	
	1.4. Jaringan konvolusional	
	1.5. Pemodelan sekuens: Jaringan berulang dan rekursif	
2	2.1. Aljabar linier	Pembelajaran mesin
	2.2. Probabilitas dan teori informasi	
	2.3. Komputasi numerik	
	2.4. Dasar-dasar pembelajaran mesin	
3	3.1. Pembelajaran tanpa pengawasan	Pembelajaran pengelompokan spektral
	3.2. Maksimasi ekspektasi	
	3.3. Prediksi terstruktur	
	3.4. Pembelajaran imitasi	

14.7 PEMBUATAN JUDUL OTOMATIS DENGAN TRANSFORMER

Dalam artikel ini, kami menyajikan pendekatan baru untuk pembuatan judul otomatis dari teks pendek menggunakan Model Bahasa Transformer yang telah dilatih sebelumnya. Model kami dapat menghasilkan judul yang sintaksis dan semantiknya benar tanpa perlu menggunakan dataset yang lebih besar untuk pelatihan. Kami memvalidasi pendekatan kami pada beberapa dataset dengan evaluasi manusia dan objektif, dan hasil yang diperoleh sangat menggembirakan. Pendekatan ini mudah diadaptasi ke berbagai domain jika tersedia cukup data. Kami juga mengusulkan pendekatan untuk menghasilkan abstrak yang baik dan efektif secara otomatis dari dokumen berukuran sedang, sehingga menciptakan alur kerja lengkap mulai dari membaca dokumen berukuran sedang hingga menghasilkan ringkasan singkat dan membuat judul untuk dokumen tersebut. Kami juga mengusulkan pendekatan pembuatan judul hierarkis untuk jalur pembelajaran dan memperoleh hasil yang menjanjikan. Ini adalah aplikasi yang sangat signifikan dalam pembelajaran yang ditingkatkan teknologi. Arah masa depan yang mungkin untuk pekerjaan kami adalah menambahkan gambaran deskriptif dan berbagai statistik jalur pembelajaran ke pratinjau dan membuat pengantar yang menarik namun singkat untuk jalur tersebut.

DAFTAR PUSTAKA

Akhil, K., Redddy, A. B., Reddy, R. R., & Borad, A. (2022, December). Semantic Approach for the classification of Knowledge representation and reasoning analysis. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)* (pp. 101-107). IEEE.

Anusha, S., Vinoth, R., & Ramshankar, N. (2026, February). Transforming Information Retrieval with AI: Semantic, Neural, and Multimodal Approaches. In *2026 International Conference on Visual Analytics and Data Visualization (ICVADV)* (pp. 416-423). IEEE.

Beheshti, A., Yang, J., Sheng, Q. Z., Benatallah, B., Casati, F., Dustdar, S., ... & Xue, S. (2023, July). ProcessGPT: transforming business process management with generative artificial intelligence. In *2023 IEEE international conference on web services (ICWS)* (pp. 731-739). IEEE.

Bhuyan, B. P., Ramdane-Cherif, A., Tomar, R., & Singh, T. P. (2024). Neuro-symbolic artificial intelligence: a survey. *Neural Computing and Applications*, *36*(21), 12809-12844.

Carbonaro, A., Ortiz-Rodriguez, F., & Janev, V. (Eds.). (2024). *Roles and Challenges of Semantic Intelligence in Healthcare Cognitive Computing*. SAGE Publications Limited.

Chaccour, C., Saad, W., Debbah, M., Han, Z., & Poor, H. V. (2024). Less data, more knowledge: Building next-generation semantic communication networks. *IEEE Communications Surveys & Tutorials*, *27*(1), 37-76.

Chamola, V., Sai, S., Sai, R., Hussain, A., & Sikdar, B. (2024). Generative AI for consumer electronics: Enhancing user experience with cognitive and semantic computing. *IEEE Consumer Electronics Magazine*, *14*(2), 10-19.

Čišić, D., Drezgić, S., Buterin, V., Gržeta, I., Kovačić, B., Poščić, P., ... & Misuraca, G. C. (2025). Semantic Mapping of AI-for-Government Research: Uncovering the Knowledge Architecture of Digital-Era Governance. *Administrative Sciences*, *16*(1), 19.

Clark, S., Battaglia, C., Castelli, I. E., Flores, E., Gold, L., Punckt, C., ... & Veit, P. (2025). Semantic resources for managing knowledge in battery research. *ChemSusChem*, *18*(16), e202500458.

Cope, B., Kalantzis, M., & Searsmith, D. (2021). Artificial intelligence for education: Knowledge and its assessment in AI-enabled learning ecologies. *Educational philosophy and theory*, *53*(12), 1229-1245.

DeBellis, M., & Neches, R. (2023). Knowledge representation and the semantic web: an historical overview of influences on emerging tools. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, *16*(6), 22-36.

Feng, C., Zhang, A., Min, G., Huang, Y., Quek, T. Q., & You, X. (2025). Towards 6G Native-AI Edge Networks: A Semantic-Aware and Agentic Intelligence Paradigm. *arXiv preprint arXiv:2512.04405*.

Ghnemat, R. (2022). Higher education transformation for artificial intelligence revolution: Transformation framework. *International Journal of Emerging Technologies in Learning (iJET)*.

Gil, Y., Garijo, D., Khider, D., Knoblock, C. A., Ratnakar, V., Osorio, M., ... & Shu, L. (2021). Artificial intelligence for modeling complex systems: taming the complexity of expert models to improve decision making. *ACM Transactions on Interactive Intelligent Systems*, *11*(2), 1-49.

Goyal, K. K. (2025, October). Semantic AI Infrastructure for Sustainable Decision Intelligence. In *2025 8th International Conference on Informatics and Computational Sciences (ICICoS)* (pp. 434-439). IEEE.

Hassan, B. A., & Rashid, T. A. (2021). Artificial intelligence algorithms for natural language processing and the semantic web ontology learning. *arXiv preprint arXiv:2108.13772*.

He, X., & Burger-Helmchen, T. (2024). Evolving knowledge management: Artificial intelligence and the dynamics of social interactions. *IEEE Engineering Management Review*, *53*(5), 215-231.

Hilger, J., Tesfaye, L., & Wahl, Z. (2026). *Bridging Knowledge, Data, and AI: Harnessing the Semantic Layer Framework to Drive Intelligence*. Springer Nature.

Hyvönen, E. (2020). Using the Semantic Web in digital humanities: Shift from data publishing to data-analysis and serendipitous knowledge discovery. *Semantic Web*, *11*(1), 187-193.

Iqbal, J., & Hera, J. (2025). Semantic-Aware Knowledge Graphs: Integrating NLP and AI for Intelligent Data Organization.

Jadala, V. C., & Burugari, V. K. (2024). Artificial intelligence techniques in semantic web services. *INTERNATIONAL JOURNAL OF INNOVATION IN MULTIDISCIPLINARY SCIENTIFIC RESEARCH*.

Jain, S., Mihindukulasooriya, N., Janev, V., & Shimizu, C. M. (2020). Semantic Intelligence,". *Artificial Intelligence of Things AIOT", Faculty Development Programme sponsored by AICTE ATAL Academy, organized by SRM Modinagar [Video]. Retrieved from https://youtu. be/Z_tbR02wsB4*.

Janev, V. (2021). Semantic intelligence in big data applications. In *Smart Connected World: Technologies and Applications Shaping the Future* (pp. 71-89). Cham: Springer International Publishing.

Khan, J. (2025). Cognitive ETL: Using AI to Interpret Complex Data Relationships. *Available at SSRN 5728984*.

Lan, Q., Wen, D., Zhang, Z., Zeng, Q., Chen, X., Popovski, P., & Huang, K. (2021). What is semantic communication? A view on conveying meaning in the era of machine intelligence. *Journal of Communications and Information Networks*, *6*(4), 336-371.

Levy, P. (2023). Semantic computing with IEML. *Collective Intelligence*, *2*(4), 26339137231207634.

Li, C., Duan, Y., Wang, L., & Tao, X. (2025). AI-enabled multimedia semantic communications. *IEEE Communications Magazine*, *63*(6), 66-73.

Lu, J., Yang, W., Xiong, Z., Xing, C., Tafazolli, R., Quek, T. Q., & Debbah, M. (2025). Generative artificial intelligence-enhanced multimodal semantic communication in internet of vehicles: System design and methodologies. *IEEE Vehicular Technology Magazine*.

Mahmood, M. A. (2025). A Hybrid Semantic AI Architecture for Real-Time Environmental Intelligence: Integrating Knowledge Graphs, Reproducible Workflows, and Adaptive Decision Systems for Earth Data Processing. *Reproducible Workflows, and Adaptive Decision Systems for Earth Data Processing (November 07, 2025)*.

Nancharaiah, B., Prasad, R. V. V. S. V., Bolla, J. V., Anakal, S., & Viji, D. (2024). Semantic Web and AI: Knowledge Representation and Reasoning. *Journal of Computational Analysis & Applications*, *33*(4).

Ogenyi, F. C., Ugwu, C. N., & Ugwu, O. P. C. (2025). A comprehensive review of AI-native 6G: integrating semantic communications, reconfigurable intelligent surfaces, and edge intelligence for next-generation connectivity. *Frontiers in Communications and Networks*, *6*, 1655410.

Okafor, A., & Brandon, S. (2025). AI-Driven Semantic Security: Encrypted Learning Models for Intelligent Knowledge Databases.

Pakgohar, A. (2025). Dynamic Computer: A Semantic Computing Architecture for Emergent Intelligence.

Petrenko, M., Cohn, E., Shchurov, O., & Malakhov, K. (2023). Ontology-driven computer systems: Elementary senses in domain knowledge processing. *South African Computer Journal*, *35*(2), 127-144.

Roy, A. (2025). Revolutionizing Digital Narratives: The Role of Semantic Web and Artificial Intelligence in Storytelling. *International Journal of Advanced Nano Computing and Analytics*, *4*(1), 14-34.

Saraswathi, M. (2024). Transforming Enterprise Operations with AI Driven Semantic Analytics and Cloud Orchestration. *International Journal of Humanities and Information Technology*, *6*(03), 70-82.

Schimanke, F., Sert, M., & Chou, B. C. H. (Eds.). (2025). *Semantic Computing And Ai: Transforming Knowledge Into Intelligence*. World Scientific.

Sheth, A., Roy, K., & Gaur, M. (2023). Neurosymbolic artificial intelligence (why, what, and how). *IEEE Intelligent Systems*, *38*(3), 56-62.

Vasilyev, N., Gromyko, V., & Anosov, S. (2020, February). Deep-learned artificial intelligence for semantic communication and data co-processing. In *Future of Information and Communication Conference* (pp. 916-926). Cham: Springer International Publishing.

Viswanathan, V., Polagani, S. S., Agarwal, R., Akula, S., Dey, S., & Kashyap, R. (2025, September). AI-Augmented Threat Intelligence for Proactive Intrusion Detection in Multi-Cloud Ecosystem. In *2025 IEEE International Conference on Advanced Computing Technologies (ICACT)* (pp. 567-572). IEEE.

Wang, Y. (2024, July). Semantic communication networks empowered artificial intelligence of things. In *2024 IEEE Annual Congress on Artificial Intelligence of Things (AIoT)* (pp. 189-193). IEEE.

Yang, Y., Zhuang, Y., & Pan, Y. (2021). Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *Frontiers of Information Technology & Electronic Engineering*, *22*(12), 1551-1558.

Yu, Y., Wang, M. H., & Zhang, J. B. (2026). Abstract Meaning Representation for Cross-Domain Knowledge Integration: A Semantic Framework for Explainable and Trustworthy AI. In *Artificial Intelligence and Human-Computer Interaction: Proceedings of the 3rd International Conference (ArtInHCI 2025), Nanning, China, 21-23 October 2025* (pp. 79-86). 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP: SAGE Publications.

KOMPUTASI SEMANTIK & AI

Transformasi Pengetahuan menuju Kecerdasan

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

BIO DATA PENULIS



Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen, ilmu sosiologi dan ilmu hukum. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik), Ilmu Perpajakan.

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7695-31-4 (PDF)
9 786347 695314