PENDEKATAN BARU EKSTRAKSI INFORMASI *BIOMEDICAL* BIG DATA *REPORT* DENGAN BIOWORDVEC MENGGUNAKAN MODEL *HYBRID LONG SHORT-TERM MEMORY – CONVOLUTIONAL NEURAL NETWORK* (LSTM – CNN)

DISERTASI

Oleh

DIAN KURNIASARI

2037061002

PROGRAM STUDI DOKTOR MIPA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

ABSTRAK

PENDEKATAN BARU EKSTRAKSI INFORMASI *BIOMEDICAL* BIG DATA *REPORT* DENGAN BIOWORDVEC MENGGUNAKAN MODEL *HYBRID LONG SHORT-TERM MEMORY – CONVOLUTIONAL NEURAL NETWORK* (LSTM – CNN)

Oleh

DIAN KURNIASARI

Peningkatan angka kematian akibat leukemia telah mendorong pesatnya pertumbuhan publikasi mengenai penyakit ini. Lonjakan publikasi tersebut berdampak signifikan pada peningkatan literatur biomedis, yang membuat ekstraksi informasi relevan tentang leukemia secara manual semakin menantang. Hal ini dikarenakan penelitian yang ada sebelumnya umumnya hanya memperhitungkan komponen leksikal dan sintaksis teks tanpa mempertimbangkan makna semantiknya.

Tujuan dilakukannya penelitian ini antara lain adalah untuk menemukan model *Deep Learning* (DL) terbaik dalam melakukan ekstraksi informasi yang relevan secara semantik pada sejumlah besar data biomedis yang disebut *biomedical* big data *report*. *Semantic Text Similarity* (STS) adalah salah satu bidang penelitian penting dalam aplikasi saat ini yang terkait dengan analisis semantik teks. Metode tersebut memungkinkan ekstraksi informasi dari suatu teks menjadi lebih bermakna karena melibatkan penerapan representasi distribusi kata-kata atau sumber eksternal pengetahuan semantik terstruktur seperti *word embedding*. Namun perlu diperhatikan bahwa *word embedding* yang digunakan harus sesuai dengan domain penelitian.

Penelitian ini mengusulkan penerapan arsitektur *Siamese Manhattan* pada model DL, yaitu model CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN, untuk melakukan analisis semantik teks biomedis. Teks biomedis yang memiliki makna semantik atau berada pada konteks yang sama direpresentasikan ke dalam bentuk vektor berdasarkan *word embedding* khusus domain biomedis, yaitu BioWordVec. Lebih lanjut, model tersebut dibangun dan dibandingkan berdasarkan jumlah lapisan tersembunyi dan metode pelabelan yang digunakan. Jumlah lapisan tersembunyi yang digunakan adalah dua dan tiga, sedangkan metode pelabelan yang digunakan adalah metode *Cosine Similarity* (CS) dan metode *Word Mover's Distance* (WMD). Hasil analisis semantik menunjukkan bahwa setiap kalimat memiliki makna semantik yang identik dengan tingkat *similarity* 1.

Hasil tersebut selanjutnya menjadi landasan untuk dilakukan klasifikasi teks sebagai bentuk aplikasi langsung dari STS. Model klasifikasi teks dibangun dan

dibandingkan berdasarkan dua skema pembagian data, yaitu *train-test split* dan *k-fold Cross Validation*. Masalah ketidakseimbangan kelas yang muncul selama proses klasifikasi kemudian diatasi melalui prosedur *resampling* menggunakan kombinasi metode *Random Undersampling* dan *Random Oversampling*. Sama seperti tahap sebelumnya yaitu STS, tahap klasifikasi teks juga menerapkan BioWordVec sebagai metode representasi kata.

Secara keseluruhan, model *hybrid* LSTM – CNN yang diusulkan untuk ekstraksi informasi memiliki performa yang lebih baik dibandingkan model CNN, LSTM, dan *hybrid* CNN – LSTM dengan nilai akurasi mencapai 100% pada tugas STS dan mencapai 99% untuk tugas klasifikasi teks. Dengan demikian, dapat disimpulkan bahwa model DL terbaik untuk melakukan ekstraksi informasi pada penelitian ini adalah model *hybrid* LSTM – CNN dengan implementasi *word embedding* khusus domain biomedis, yaitu BioWordVec.

Kata Kunci: Klasifikasi Teks, *Semantic Text Similarity*, BioWordVec, *Hybrid* LSTM – CNN.

ABSTRACT

NEW APPROACH TO BIOMEDICAL BIG DATA REPORT INFORMATION EXTRACTION WITH BIOWORDVEC USING HYBRID LONG SHORT-TERM MEMORY – CONVOLUTIONAL NEURAL NETWORK (LSTM – CNN) MODEL

By

DIAN KURNIASARI

The rise in death rates associated with leukemia has fueled the rapid expansion of publications focused on this disease. The rise in the number of publications has substantially affected the growth of biomedical literature, making it more challenging to manually extract pertinent information concerning leukemia. That is because prior studies typically focused solely on the lexical and syntactic aspects of the text, neglecting its semantic significance.

This study aims to identify the most effective Deep Learning (DL) model for extracting semantically significant information from a substantial volume of biomedical data called the Biomedical Big Data Report. Semantic Text Similarity (STS) is a crucial field of research in contemporary applications that deal with the semantic analysis of texts. This approach enhances extracting information from a text by utilizing a distributed model of words or an external source of organized semantic knowledge, such as word embedding. Nevertheless, it is essential to acknowledge that word embedding must suit the specific research field.

This study suggests implementing the Siamese Manhattan architecture in Deep Learning (DL) models, namely Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, hybrid CNN-LSTM models, and hybrid LSTM-CNN models, to do semantic analysis on biomedical text. Biomedical language with semantic significance or in the same context is transformed into vector representation using word embedding techniques specifically designed for the biomedical field, known as BioWordVec. In addition, the models are constructed and evaluated based on the number of hidden layers and labelling techniques employed. Two to three hidden layers are utilised, along with the Cosine Similarity (CS) and Word Mover's Distance (WMD) tagging methods. The findings of the semantic analysis indicate that every sentence has the same semantic meaning, with a similarity level of 1.

These results are the foundation for text classification, which directly implements STS. Text classification models are constructed and evaluated using two data partitioning methods: train-test split and k-fold Cross Validation. The class imbalance issue during the classification process is addressed using a resampling technique that combines Random Undersampling and Random Oversampling

approaches. Like the previous step, STS, the text categorization stage utilizes BioWordVec to represent words.

The hybrid LSTM – CNN model outperforms the CNN, LSTM, and hybrid CNN – LSTM models in information extraction, achieving accuracy rates of 100% on the STS task and 99% on the text classification task. Thus, it can be concluded that the best DL model for extracting information in this research is a hybrid LSTM – CNN model with the implementation of word embedding specifically for the biomedical domain, namely BioWordVec.

Keywords: Text Classification, Semantic Text Similarity, BioWordVec, Hybrid LSTM – CNN.

PENDEKATAN BARU EKSTRAKSI INFORMASI *BIOMEDICAL* BIG DATA *REPORT* DENGAN BIOWORDVEC MENGGUNAKAN MODEL *HYBRID LONG SHORT-TERM MEMORY – CONVOLUTIONAL NEURAL NETWORK* (LSTM – CNN)

Disertasi untuk memperoleh gelar Doktor dalam ilmu MIPA
Universitas Lampung

Dipertahankan di hadapan
Dewan Penguji Program Pascasarjana
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung

Oleh

DIAN KURNIASARI

2037061002

PROGRAM STUDI DOKTOR MIPA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

MENYETUJUI

Judul Disertasi	:	Pendekatan Baru Ekstraksi Informasi *Biomedical* Big Data *Report* dengan BioWordVec menggunakan Model *Hybrid Long Short-Term Memory – Convolutional Neural Network* (LSTM – CNN)
Nama Mahasiswa	:	Dian Kurniasari
Nomor Pokok Mahasiswa	:	2037061002
Program Studi	:	Doktor MIPA
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

Komisi Promotor

Promotor,

Prof. Drs. Mustofa Usman, M.A., Ph.D.
NIP 195701011984031020

Ko-Promotor 1

Ir. Warsono, M.S., Ph.D.
NIP 196302161987031003

Ko-Promotor 2

Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D.
NIP 198301102008121002

Direktur Program Pascasarjana

Prof. Dr. Ir. Murhadi, M.Si.
NIP. 196403261989021001

Ketua Program Studi Doktor MIPA,

Dr. G. Nugroho Susanto, M.Sc.
NIP 196103111988031001

PENGESAHAN PENGUJI

PENDEKATAN BARU EKSTRAKSI INFORMASI *BIOMEDICAL* BIG DATA *REPORT* DENGAN BIOWORDVEC MENGGUNAKAN MODEL *HYBRID LONG SHORT- TERM MEMORY – CONVOLUTIONAL NEURAL NETWORK* (LSTM – CNN)

DISERTASI

OLEH

DIAN KURNIASARI
NPM 2037061002

Tim Penguji

Jabatan	Nama	Tanda Tangan
Ketua	Dr. Eng. Heri Satria, S.Si., M.Si. NIP 197110012005011002	
Sekretaris 1	Dr. Muslim Ansori, S.Si., M.Si. NIP 197202271998021001	
Sekretaris 2	Dr. G. Nugroho Susanto, M.Sc. NIP 196103111988031001	
Anggota	Prof. Drs. Mustofa Usman, M.A., Ph.D. NIP 195701011984031020	
	Ir. Warsono, M.S., Ph.D. NIP 196302161987031003	
	Favorisen R. Lumbanraja, S.Kom., M.Si., Ph.D. NIP 198301102008121002	
	Dr. rer. Nat. Akmal Junaidi, M.Sc. NIP 197101291997021001	
	Dr. dr. Betta Kurniawan, M.Kes. NIP 197810092005011001	
	Prof. Dr. Ir Agus Buono, M.Si, M.Kom NIP 196607021993021001	

Telah dipertahankan di depan tim penguji pada sidang terbuka dinyatakan telah memenuhi syarat pada tanggal 22 Maret 2024

Dekan FMIPA Unila

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP 197110012005011002

Direktur Pascasarjana

Prof. Dr. Ir. Murhadi, M.Si
NIP 196403261989021001

LEMBAR IDENTITAS DAN PENGESAHAN

A.	Judul Disertasi	:	Pendekatan Baru Ekstraksi Informasi *Biomedical* Big Data *Report* dengan BioWordVec menggunakan Model *Hybrid Long Short-Term Memory – Convolutional Neural Network* (LSTM – CNN)
B.	Bidang Ilmu	:	Statistika dan Science Data
	1. Ketua		
	a. Nama	:	Dian Kurniasari
	b. Gol/Pangkat/NPM	:	IV.a/ Pembina/2037061002
	c. Jabatan Fungsional	:	Lektor Kepala
	d. Fakultas/Bagian	:	Matematika dan Ilmu Pengetahuan Alam/ Statistika dan Science Data
	e. Universitas	:	Lampung
	2. Anggota Peneliti	:	3 (tiga) orang
	3. Lokasi Penelitian	:	Kota Bandar Lampung

Bandar Lampung, 22 Maret 2024

Mengetahui,
Dekan FMIPA Unila

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Peneliti

Dian Kurniasari
NPM. 2037061002

DAFTAR RIWAYAT HIDUP

Nama : Dian Kurniasari

Tanggal Lahir : Mentok – Bangka, 05 Maret 1969

Jabatan : Dosen Jurusan Matematika di Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung

Pendidikan : S1 Statistika Universitas Gadjah Mada
S2 Applied Mathematics Curtin University of Technology

Prestasi di Bidang Tri Dharma Perguruan Tinggi

1. Perolehan Hibah Penelitian Selama Lima Tahun Terakhir (2018 – 2023)

- Eksplorasi Simulasi *Monte Carlo* dalam Metode Momen, Metode Peluang Terboboti dan Metode Kemungkinan Maksimum pada Distribusi *Generalized Pareto* (2018).
- Metode *Empirical Best Linear Unbiased Prediction* (EBLUP) dan *Spatial Empirical Best Linear Unbiased Prediction* (SEBLUP) dalam Pendugaan Area Kecil (2019).
- Perbandingan Metode Pendugaan Parameter pada Distribusi *Generalized Beta* 2 (2019).
- Karakteristik Penduga *Bayes* Distribusi Binomial dengan *Prior Beta* (2020)
- Analisis Dinamik Struktur dengan Model *Vector Autoregressive* X Data Jumlah Uang Beredar dan Inflasi Indonesia (2020).
- *Deep Learning* dengan Model BioWordVec untuk Klasifikasi Laporan Medikal (2023).
- Perbandingan Pembelajaran Kesamaan Teks menggunakan *Deep Siamese Manhattan* antara *Cosine Similarity* dan *Word Mover's Distance* pada Laporan Medikal (2023).

2. Luaran Penelitian yang dihasilkan dari Disertasi

- Hak Cipta untuk Program Komputer dengan Judul Ciptaan "Perbandingan Pembelajaran Kesamaan Teks menggunakan *Deep Siamese Manhattan* antara *Cosine Similarity* dan *Word Mover's Distance* pada Laporan Medikal".
- LSTM-CNN Hybrid Model Performance Improvement with BioWordVec for Biomedical Report Big Data Classification (*Accepted* pada Jurnal *Science and Technology Indonesia*).
- Hak Cipta untuk Program Komputer dengan Judul Ciptaan "*Deep Learning* dengan Model BioWordVec untuk Klasifikasi Laporan Medikal".
- Comparison of Learning Text Similarity using Deep Siamese Manhattan between Cosine Similarity and Word Mover's Distance in Medical Reports (*Review Round* I pada Jurnal *International Journal of Electrical and Computer Engineering*).
- Deep learning with BioWordVec model for classification of medical reports (*Under Review* pada Jurnal *International Journal of Advances in Intelligent Informatics*).

SANWACANA

Puji syukur penulis ucapkan kehadirat Allah, karena atas rahmat dan hidayah-Nya disertasi ini dapat diselesaikan. Disertasi dengan judul "Pendekatan Baru Ekstraksi Informasi *Biomedical* Big Data *Report* dengan BioWordVec menggunakan Model *Hybrid Long Short-Term Memory – Convolutional Neural Network* (LSTM – CNN)" adalah salah satu syarat untuk memperoleh gelar Doktor di Fakultas MIPA Universitas Lampung.

Pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., selaku Rektor Universitas Lampung;
2. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung;
3. Bapak Dr. G. Nugroho Susanto, M.Sc selaku Ketua Program Studi Doktor MIPA Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung;
4. Bapak Prof. Drs. Mustofa Usman, M.A., Ph.D. selaku Promotor dan Pembimbing Akademik yang telah membimbing hingga disertasi ini selesai;
5. Bapak Ir. Warsono, M.S., Ph.D. selaku Co-Promotor I yang telah membantu dan membimbing dalam proses penyusunan disertasi;
6. Bapak Favorisen Rosyking Lumbanraja S.Kom., M.Sc., Ph.D. selaku Co-Promotor II yang telah banyak membantu, membimbing, memberikan arahan dan saran sehingga disertasi ini dapat diselesaikan dengan baik;
7. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku Penguji atas semua masukkan dan saran yang diberikan supaya disertasi ini menjadi lebih baik;
8. Bapak Dr. dr. Betta Kurniawan, M.Kes. selaku Penguji atas semua masukkan dan saran yang diberikan supaya disertasi ini menjadi lebih baik;

9. Suamiku "Syamsul Hilal" tercinta yang telah membersamai dalam suka dan duka, memberikan restu, mengirimkan do'a, serta dukungan agar disertasi ini dapat diselesaikan dengan baik;
10. Dr. Aang Nuryaman, dan Dr. Ahmad Faisol yang telah banyak memberikan bantuan, nasihat, dan semangat selama proses penyusunan disertasi;
11. Prof. Wamiliana, Prof. Asmiati, Widiarti, M.Si., Dr. Fitriani, dan Dr. Notiragayu yang telah banyak memberikan bantuan, nasihat, dan semangat selama proses penyusunan disertasi;
12. Ibuku tercinta yang telah memberikan restu, mengirimkan do'a, serta dukungan agar disertasi ini dapat diselesaikan dengan baik;
13. Kakakku dan adik-adikku tercinta yang telah memberikan restu, mengirimkan do'a, serta dukungan agar disertasi ini dapat diselesaikan dengan baik;
14. Sahabat seperjuangan, Bernadhita Herindri Samodera Utami atas kebersamaan dan dukungannya selama ini;
15. Keluarga besar Doktor MIPA 2020: Aristoteles, Rico, Syaiful, Ridho, Arif, Nurjoko, Reni, dan Tika atas kebersamaannya selama ini;
16. Seluruh civitas akademika UNILA yang telah memberikan bantuan dan tidak dapat saya sebutkan satu persatu, semoga Allah membalas amal kebaikan kita semua.

Semoga disertasi ini dapat memberikan manfaat bagi kita semua dalam rangka pengembangan ilmu pengetahuan dan teknologi untuk masa depan, Aamiin.

Bandar Lampung, 22 Maret 2024

DIAN KURNIASARI
NPM. 2037061002

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR PERSAMAAN

DAFTAR KODE PROGRAM

DAFTAR SINGKATAN

AI	: *Artificial Intelligence*
ANN	: *Artificial Neural Network*
BioNLP	: *Biomedical Natural Language Processing*
BOW	: *Bag of Words*
CART	: *Classification and Regression Tree*
CM	: *Confusion Matrix*
CMeSH	: *Chinese Medical Subject Headings*
CNN	: *Convolutional Neural Network*
CS	: *Cosine Similarity*
DDI	: *Drug-Drug Interaction*
DL	: *Deep Learning*
ER	: *Extraction Relation*
FN	: *False Negative*
FP	: *False Positive*
GRU	: *Gated Recurrent Unit*
HPC	: *High Performance Computing*
IE	: *Information Extraction*
IMDB	: *Internet Movie Database*
IR	: *Information Retrieval*
KG	: *Knowledge Graph*
LCQMC	: *Large-scale Chinese Question Matching Corpus*
LIME	: *Local Interpretable Model-Agnostic Explanations*
LDA	: *Latent Dirichlet Allocation*
LSA	: *Latent Semantic Analysis*
LSTM	: *Long Short-Term Memory*
MeSH	: *Medical Subject Headings*

MCNN-LSTM	: *Multiclass Convolutional Neural Network – Long Short-Term Memory*
ML	: *Machine Learning*
MLP	: *Multi Layer Perceptron*
MSR	: *Microsoft Research*
NB	: *Naïve Bayes*
NCBI	: *National Center for Biotechnology Information*
NER	: *Named Entity Recognition*
NLM	: *National Library of Medicine*
NLP	: *Natural Language Processing*
OS	: *Operating System*
PMC	: *PubMed Central*
POS	: *Part-Of-Speech tagging*
PPV	: *Positive Predictive Value* atau Presisi
RA	: *Rheumatoid Arthritis*
RNN	: *Recurrent Neural Network*
ReLU	: *Rectified Linear Unit*
SMOTE	: *Synthetic Minority Oversampling Technique*
SRL	: *Semantic Role Labelling*
SVM	: *Support Vector Machine*
STS	: *Semantic Textual Similarity*
TF-IDF	: *Term Frequency – Inverse Document Frequency*
Tanh	: *Tangent Hyperbolic*
TM	: *Text Mining*
TN	: *True Negative*
TP	: *True Positive*
TPR	: *True Positive Rate* atau *Recall*
UMLS	: *Unified Medical Language System*
WMD	: *Word Mover's Distance*

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Leukemia atau kanker darah adalah penyakit yang disebabkan oleh rusaknya sistem kekebalan tubuh akibat adanya peningkatan jumlah sel darah putih dalam darah atau sumsum tulang (Ahmed dkk., 2019; Du dkk., 2022; Saeed dkk., 2022). Secara global, angka kematian akibat penyakit leukemia terus meningkat selama beberapa dekade, dari 263,263 kasus kematian pada tahun 1990 menjadi 334,592 kasus kematian pada tahun 2019 (Sharma dan Jani, 2022). Pada tahun 2020, data dari proyek kanker global (GLOBOCAN) menyatakan bahwa leukemia berada pada urutan ke-11 sebagai penyebab utama kematian akibat kanker, dengan 311,594 kasus kematian karena leukemia. GLOBOCAN juga menyatakan bahwa leukemia adalah kanker yang paling umum terjadi dan menempati urutan ke-15, dengan lebih dari 470,000 kasus baru dilaporkan pada tahun 2020 (Rifat dkk., 2023). Oleh karena itu, akibat dari banyaknya kasus baru yang dilaporkan dan angka kematian yang terus meningkat, leukemia telah menjadi salah satu topik yang paling banyak diteliti dalam bidang biomedis (Sharma dkk., 2018).

Literatur penelitian mengenai leukemia dapat diakses melalui berbagai *database* ilmiah seperti PubMed, ScienceDirect, Scopus dan Google Scholar. Meskipun demikian, Tober (2011) dan Masic (2012) menyatakan bahwa di antara keempat *database* tersebut, PubMed merupakan *database* terbesar dan paling populer untuk domain biomedis karena mampu menyajikan hasil pencarian dokumen biomedis secara lebih komprehensif dibandingkan *database* lainnya. PubMed merupakan *database* biomedis gratis yang dikembangkan oleh *National Center for Biotechnology Information* (NCBI), sebuah divisi dari *National Library of Medicine* (NLM) yang berisi sejumlah besar literatur biomedis seperti jurnal ilmiah,

catatan klinis, laporan medikal dan data tidak terstruktur lainnya. Karena aksesnya yang mudah, efisien, dan ramah pengguna, PubMed telah menjadi referensi utama bagi para peneliti biomedis di seluruh dunia (Zhao dkk., 2022; Parwez dkk., 2023).

Dokumentasi awal penelitian leukemia di PubMed dimulai pada tahun 1864 oleh Weeks (1864). Namun, pada Februari 2024, apabila istilah "leukemia" dicari di PubMed, maka akan ditemukan 374,286 publikasi yang menunjukkan pesatnya pertumbuhan publikasi terkait leukemia. Tingkat publikasi yang cepat tersebut berdampak pada peningkatan jumlah data teks biomedis yang signifikan, sehingga diperlukan suatu metode yang mampu mengubah literatur biomedis tersebut menjadi informasi yang bermanfaat dan relevan. Prosedur standar untuk menganalisis teks biomedis tersebut adalah dengan melakukan meta analisis secara komprehensif, terintegrasi dan sistematis pada kumpulan data tekstual. Hasil dari analisis ini sangat akurat karena dilakukan secara manual oleh ekspert di bidang biomedis. Namun metode ini memiliki beberapa kelemahan, yaitu membutuhkan lebih banyak *effort*, waktu yang lama, dan kekurangan ekspert pada bidang biomedis yang signifikan. Akibatnya, diperlukan metode alternatif yang lebih efektif dan efisien dalam menganalisis dan mengekstraksi teks biomedis untuk mengatasi pertumbuhan publikasi yang terus meningkat (Paraskevopoulos dkk., 2022).

Metode alternatif yang dapat digunakan dan terbukti lebih efisien untuk ekstraksi informasi yang bermanfaat dan relevan adalah program komputer dengan *High Performance Computing* (HPC) yang mencakup *processor* dan ukuran RAM yang besar. Menurut Stanojevic dkk. (2021) dan Usman dkk. (2023), HPC lebih efisien dan membutuhkan waktu yang lebih singkat karena menggunakan *Machine Learning* (ML) dan metode kecerdasan buatan lainnya dalam memecahkan masalah kompleksitas analisis teks.

Metode kecerdasan buatan yang memungkinkan program komputer untuk mengekstrak informasi berkualitas tinggi dari "bahasa alami" yang tersimpan dalam format semi terstruktur dan tidak terstruktur seperti teks disebut *text mining*

(Nellimella dkk., 2022). Pemaknaan "bahasa alami" pada dasarnya mencakup tingkat leksikal, sintaksis, dan semantik. Di mana setiap tingkatnya semakin kompleks dan memerlukan pengolahan yang lebih canggih dibandingkan tingkat sebelumnya (Sinoara dkk., 2017). Sehingga, *text mining* biasanya melibatkan penggunaan suatu metodologi yang dikenal sebagai *Natural Language Processing* (NLP). *Natural Language Processing* merupakan suatu pendekatan yang dapat mengubah data tidak terstruktur, seperti teks, menjadi informasi terstruktur yang dapat dipahami oleh program komputer (Gu dkk., 2022).

Akan tetapi, agar teknik NLP berhasil diterapkan pada data teks dalam domain biomedis, diperlukan sumber daya berupa korpora beranotasi yang dirancang khusus untuk NLP dalam domain tersebut. Salah satu korpora beranotasi dalam domain biomedis yang dibangun berdasarkan literatur penyakit leukemia adalah GENIA *Biomedical Event*. GENIA *Biomedical Event* merupakan korpora yang dibangun dan dianotasi secara manual berdasarkan "kata pemicu" dan "peristiwa biomedis" yang berkaitan dengan penyakit leukemia oleh Kim dkk. (2008). Korpora tersebut terdiri dari 1,000 abstrak PubMed dengan lebih dari 8,000 kalimat yang relevan terhadap reaksi biologis mengenai faktor transkripsi pada sel darah manusia dan leukemia karena dipilih berdasarkan *query* PubMed untuk tiga istilah *Medical Subject Headings* (MeSH), yaitu "manusia", "sel darah", serta "faktor transkripsi" (Björne dkk., 2010). *Medical Subject Headings* adalah istilah standar dan sangat spesifik dalam domain biomedis yang bertujuan untuk mengindeks setiap artikel di PubMed (Sriganesh, 2011). Oleh karena itu, data GENIA *Biomedical Event* memiliki banyak informasi yang dapat digunakan untuk perkembangan di bidang medis, khususnya leukemia.

Menurut Vayena dan Blasimme (2017), semua data yang relevan dengan bidang medis atau kesehatan, serta dapat digunakan untuk tujuan penambangan data prediktif yang berhubungan dengan kesehatan disebut *biomedical* big data. Jenis data ini dihasilkan oleh layanan kesehatan, kegiatan kesehatan masyarakat, penelitian biomedis, *Electronical Health Record*, *National Healthcare Data*, dan sebagainya. Berdasarkan definisi tersebut, GENIA *Biomedical Event* dapat

dikategorikan sebagai *biomedical* big data. Istilah "big data" yang disematkan mengacu pada karakteristik kumpulan data yang memiliki ukuran besar, lebih bervariasi, dan kompleks dengan kesulitan menyimpan, menganalisis, dan memvisualisasikan untuk proses atau hasil lebih lanjut (Yaseen dan Obaid, 2020).

Secara tradisional, teknik *text mining* dan NLP mengandalkan kombinasi representasi *bag-of-words* dan *data mining*. Sayangnya, teknik tersebut hanya memperhitungkan komponen leksikal dan sintaksis teks tanpa mempertimbangkan makna semantiknya. Dengan demikian, pada penelitian sebelumnya, komponen leksikal dan sintaksis telah banyak dieksplorasi dalam *text mining* dibandingkan dengan komponen semantik, namun belum memberikan hasil ekstraksi informasi yang berarti (Aggarwal dkk., 2012). Kondisi ini menyebabkan para peneliti mulai beralih ke analisis semantik teks untuk menemukan hasil penambangan teks yang lebih baik. Selain itu, meningkatnya minat ini dapat dikaitkan dengan kemajuan dalam kapasitas komputasi, dan pengembangan dalam pemrosesan bahasa alami, yang memungkinkan ekstraksi informasi dari suatu teks menjadi lebih bermakna (Sinoara dkk., 2017).

Semantic Text Similarity (STS) adalah salah satu bidang penelitian penting dalam aplikasi saat ini yang terkait dengan analisis semantik teks. Tugas STS pertama kali diterapkan dalam temu kembali informasi, seperti pencarian *web*, *subtopic mining*, *word-sense disambiguation*, dan *relevance feedback* dengan menggunakan model ruang vektor (Shajalal dan Aono, 2019). Model tersebut digunakan untuk menghitung kesamaan semantik antara *query* dan dokumen dengan tujuan memperoleh dokumen yang paling relevan dengan *query* tertentu (Metzler dkk., 2007; Li dan Xu, 2013; Shajalal dkk., 2016).

Sedangkan pada NLP, STS merupakan tugas klasik yang bertujuan untuk menentukan tingkat kesamaan semantik antara dua kata atau kalimat dalam teks dan mengevaluasi sejauh mana dua kalimat tersebut identik satu sama lain (Reimers dkk., 2016; Wang dkk., 2020). *Semantic Textual Similarity* memainkan peran penting dalam berbagai tugas NLP, termasuk BioNLP, karena dapat mencegah

penilaian yang tidak akurat dan kesalahpahaman istilah atau kalimat dalam mengekstraksi informasi dari dokumen biomedis (Alam dkk., 2020).

Kesamaan semantik yang dimaksud dalam NLP adalah kesamaan yang didasarkan pada makna kata dalam linguistik, yang mencakup kesamaan kata ke kata berdasarkan representasi distribusi kata-kata atau sumber eksternal pengetahuan semantik terstruktur seperti *word embedding* (Nguyen dkk., 2019). *Word embedding* memiliki peran penting pada tugas STS karena setiap kata pada *word embedding* adalah vektor yang merepresentasikan sebuah titik pada ruang dengan dimensi tertentu. Dengan demikian, kata-kata yang berada pada konteks yang sama, atau memiliki makna semantik yang sama akan berada tidak jauh satu sama lain pada ruang tersebut. Implementasi *word embedding* untuk mengukur kesamaan semantik dalam teks telah banyak dilakukan dan ditemukan pada berbagai penelitian terkait *text summarization, machine translation, paraphrase detection,* dan analisis sentimen di berbagai domain (Aliguliyev, 2009; Kenter dan De Rijke, 2015; Farouk, 2019).

Metode berbasis korpora, berbasis pengetahuan, dan berbasis karakter adalah tiga pendekatan utama yang sering digunakan untuk menentukan STS pada penelitian sebelumnya. Menurut penelitian yang dilakukan oleh Sunilkumar dan Shaji (2019), di antara ketiga metode tersebut, metode berbasis korpora merupakan metode yang memberikan hasil lebih baik. Metode berbasis korpora memiliki dua pendekatan yang berbeda secara statistik. Pertama, pendekatan yang melibatkan penggunaan teknik analisis statistik tradisional seperti *Latent Semantic Analysis* (LSA) (Kashyap dkk., 2016), *Latent Dirichlet Allocation* (LDA) (Niraula dkk., 2013), *Support Vector Machine* (SVM) (Béchara dkk., 2015), dan *Random Forest* (Buscaldi dkk., 2015). Pendekatan ini memiliki algoritma yang sederhana namun tidak fokus pada penggalian informasi semantik yang mendalam. Pendekatan kedua melibatkan penggunaan *Deep Learning* (DL) untuk mengekstrak representasi fitur kata pada *word embedding* dengan menangkap konteks spesifik dalam teks dan menentukan kesamaan label dengan *multilayer perceptron* (Zhao dkk., 2022).

Namun, penerapan metode berbasis korpora dalam penelitian NLP biomedis masih terbatas karena membutuhkan korpora khusus domain biomedis atau sumber pengetahuan biomedis (Yang dkk., 2020). Selain itu, proses anotasi atau pelabelan di bidang biomedis memerlukan keahlian profesional medis (Romanov dan Shivade, 2018; Wang dkk., 2020). Akibatnya, ketersediaan kumpulan data yang memiliki label terbatas dan penelitian terkait STS biomedis terhambat.

Metode yang paling mudah dan umum digunakan dalam menentukan STS pada data yang tidak memiliki label adalah dengan menghitung *Cosine Similarity* (CS) dan *Word Mover's Distance* (WMD) antara dua kalimat. Namun, kedua metode tersebut cenderung memiliki kinerja yang rendah dan kesulitan membedakan kalimat dengan istilah yang sama meskipun memiliki makna semantik yang berbeda (Brunila dan LaViolette, 2021; Wang, dkk., 2020; Yamagiwa dkk., 2022).

Pencapaian terbaru tugas STS berhasil dicapai oleh model CNN dan LSTM dengan cara melakukan analisis dan pembelajaran yang mendalam pada kata dan kalimat untuk mempertimbangkan makna dan struktur kalimat untuk menghitung STS (Pontes dkk., 2018). *Convolutional Neural Network* merupakan salah satu jenis algoritma DL yang memungkinkan identifikasi pola dan hubungan semantik antara kata-kata dalam sebuah kalimat menggunakan lapisan konvolusi yang terhubung ke fitur lokal. Model CNN telah terbukti berhasil dalam berbagai tugas NLP, seperti representasi kalimat, pencarian *query retrieval*, dan *semantic parsing*. Selain itu, CNN juga terbukti unggul dalam mengidentifikasi kemiripan semantik antara pasangan teks, sehingga menunjukkan kinerja yang sangat baik dalam tugas yang berkaitan dengan STS (Zheng dkk., 2019). Namun, arsitektur CNN tidak dapat mengeksplorasi hubungan urutan kata dalam kalimat karena CNN tidak memperhitungkan urutan kata dalam analisisnya. Berbeda dengan CNN, arsitektur LSTM secara khusus dirancang untuk memproses data sekuensial sehingga model ini sangat cocok untuk menangani tugas yang melibatkan kesamaan semantik karena mampu memahami konteks dan makna kata dalam kalimat.

Metode lain yang mungkin menjadi pilihan terbaik dalam kategori ini adalah menggunakan arsitektur *Siamese Neural Network* yang diusulkan oleh Mueller dan Thyagarajan (2016). Mueller dan Thyagarajan (2016) mengusulkan penerapan *Siamese Neural Network* pada model *Long Short-Term Memory* (LSTM) dengan menggunakan *pre-trained word embedding* bernama Word2Vec. Kemudian metrik kesamaan pada keluaran akhir dihitung dengan menggunakan jarak *Manhattan*. Oleh karena itu, model tersebut dikenal dengan nama *Siamese Neural Network – Manhattan* atau secara singkat disebut model *Siamese – Manhattan*. *Siamese – Manhattan* mengacu pada fakta bahwa metrik kesamaan yang digunakan untuk menentukan kesamaan pada keluaran akhir model adalah jarak *Manhattan*, bukan metrik lainnya seperti jarak *Cosine* atau *Euclidean* (Othman dkk., 2019).

Pada dasarnya, arsitektur *Siamese Neural Network* terdiri dari dua jaringan saraf identik yang bekerja secara paralel untuk mengekstraksi representasi kata dari vektor masukkan. Sehingga penerapan arsitektur tersebut memungkinkan model untuk mengkonfigurasi jaringan sesuai dengan tugas, termasuk penggunaan model DL seperti LSTM (de Souza dkk., 2020). Metrik kesamaan pada lapisan keluaran kemudian dapat dihitung berdasarkan metrik jarak, seperti *Manhattan*, *Euclidean* dan *Cosine*. Namun, penelitian oleh (Ranasinghe dkk., 2019; Shi dkk., 2020; Chicco, 2021) menunjukkan bahwa metrik kesamaan *Manhattan* merupakan teknik terbaik untuk mencapai konvergensi yang lebih cepat dan akurasi yang lebih tinggi dibandingkan dengan metrik kesamaan lainnya, termasuk jarak *Cosine*.

Selain model LSTM, arsitektur *Siamese Neural Network* juga dapat diterapkan pada model *Convolutional Neural Network* (CNN) seperti penelitian yang dilakukan oleh Yang dkk. (2023). Pada penelitian tersebut, *Siamese Neural Network* – CNN digunakan untuk memanfaatkan informasi dari data multimodal guna memprediksi peristiwa terkait *Drug – Drug Interaction* (DDI). *Siamese Neural Network* – CNN menganggap setiap obat secara terpisah sebagai masukkan pada dua *sub-network* CNN dari jaringan *Siamese*, di mana kedua CNN berbagi parameter dan mempelajari informasi multimodal obat secara individual, dan kemudian menggabungkan representasi fitur dan memasukkannya ke dalam

perceptron multilayer untuk mendapatkan prediksi keluaran dari kategori peristiwa DDI.

Identifikasi STS pada domain biomedis mempunyai banyak aplikasi langsung dalam ekstraksi informasi, misalnya pada pencarian dan klasifikasi kalimat biomedis, serta aplikasi tidak langsung, seperti menjawab pertanyaan biomedis dan pelabelan dokumen biomedis (Chen dkk., 2020). Peran STS menjadi sangat penting dalam aplikasi-aplikasi tersebut dikarenakan banyak kata-kata biomedis yang mempunyai arti berbeda apabila konteks teksnya berbeda (Li dkk., 2022). Mengingat akan hal tersebut, maka penting untuk menganalisis hasil identifikasi STS biomedis secara menyeluruh melalui aplikasi-aplikasinya, baik secara langsung maupun tidak langsung.

Aplikasi-aplikasi STS dapat dianalisis dengan beberapa metode, salah satunya dengan menggunakan klasifikasi. Klasifikasi adalah teknik *data mining* yang digunakan untuk memprediksi kategori dari objek yang belum memiliki kategori (Ha dkk., 2011). Masalah yang sering muncul dalam klasifikasi adalah data yang tidak seimbang (*imbalanced data*) pada kumpulan data. Kondisi tersebut menyebabkan proses klasifikasi menjadi kurang akurat, karena data yang seharusnya masuk ke dalam kelas minoritas terklasifikasi pada kelas mayoritas (Shaikh dkk., 2021). Tingkat kerumitan masalah ketidakseimbangan data dipicu oleh beberapa hal, seperti kompleksitas tugas, rasio ketidakseimbangan data, dan jumlah data yang digunakan pada proses pelatihan (L'Heureux dkk., 2017). Semakin tinggi tingkat ketidakseimbangan data, semakin bias terhadap kelas mayoritas (Liu dkk., 2019).

Klasifikasi yang andal pada kasus ketidakseimbangan data merupakan topik penelitian yang penting (Johnson dan Khoshgoftaar, 2019), karena ketidakseimbangan ekstrim secara alami akan berimplikasi terhadap keterlibatannya pada banyak aplikasi dunia nyata, seperti deteksi penipuan dan klasifikasi teks. Oleh karena itu, kondisi data yang tidak seimbang perlu diatasi sebelum melakukan klasifikasi. Terdapat beberapa pendekatan yang dapat

dilakukan untuk mengatasi ketidakseimbangan data. Salah satunya adalah pendekatan data dengan cara mengurangi (*undersampling*) data, menambahkan (*oversampling*) data, atau melakukan kombinasi antara *undersampling* dan *oversampling* data.

Klasifikasi teks juga merupakan tugas klasik dan mendasar di bidang NLP (Li dan Gong, 2021; Minaee dkk., 2021), sehingga penerapannya dapat ditemukan pada berbagai tugas NLP (Li dkk., 2022), seperti *sentiment analysis*, penjawab otomatis (*question answering*), *dialog act classification*, dan *topic labeling*. Penelitian mengenai klasifikasi teks juga telah banyak dilakukan dengan melibatkan berbagai algoritma tradisional ML seperti *k-Nearest Neighbors* (Nikhath dkk., 2016) , *Naive Bayes* (Di dan Duan, 2014; Xu, 2018), *Support Vector Machine* (Goudjil dkk., 2018), *Logistic Regression* (Shah dkk., 2020), *Maximum Entropy models*, *Support Vector Machine* (SVM), dan sebagainya (Zong dkk., 2021). Sayangnya, implementasi ML tradisional untuk klasifikasi teks perlu ditingkatkan karena masih terdapat informasi semantik teks yang abstrak dan relevansi konteks yang kuat.

Metode NLP berupa ML tradisional juga memiliki kelemahan, yaitu performa model yang dihasilkan lebih rendah dibandingkan dengan performa model DL berbasis *neural network* dalam hal pelatihan pada kumpulan data skala besar (Kim, 2014; Kowsari dkk., 2017; Gao dkk., 2019). Meskipun demikian, klasifikasi teks menggunakan model ML maupun model DL berbasis *neural network* sangat tergantung pada efektivitas *word embedding* sebagai salah satu masukkan utamanya seperti yang dilakukan oleh (Bengio dkk., 2003; Collobert dan Weston 2008; Turian dkk., 2010) untuk model ML, dan (Collobert dkk. 2011; Tang dkk., 2014; Ganguly dkk., 2015; Wieting dkk., 2016; Arora dkk., 2017; Kilimci dan Akyokus, 2018) untuk model DL.

Word embedding merupakan salah satu metode representasi kata, di mana setiap kata dalam teks direpresentasikan ke dalam ruang vektor yang telah ditentukan (Almeida dan Xexéo, 2019). Posisi kata tersebut dipelajari berdasarkan kata di dekatnya yang memungkinkan model untuk menangkap makna semantik dan

sintaktik kata (Li dan Gong, 2021). Pada awal kemunculannya, *word embedding* umumnya diperoleh dari data yang dilatih untuk tugas spesifik, atau disebut *custom-trained embedding*. Kemudian, pada tahun 2013 (Mikolov, dkk., 2013a; Mikolov, dkk., 2013b) memperkenalkan pendekatan baru untuk memperoleh *word embedding* dengan cara melatih data dalam jumlah besar pada korpora dengan domain tertentu yang sering disebut sebagai *pre-trained word embedding.*

Pada *custom-trained embedding*, vektor representasi kata yang dihasilkan berasal dari himpunan data pelatihan sehingga semantik korpora pelatihan dapat diwakili dengan lebih baik. Di sisi lain, *pre-trained word embedding* menghasilkan vektor representasi kata yang diperoleh dari pelatihan vektor kata pada kumpulan data besar, kemudian disimpan, dan selanjutnya digunakan untuk menyelesaikan tugas-tugas lainnya. Perbedaan mendasar antara *pre-trained word embedding* dengan *custom-trained embedding* adalah metode ini mampu menangkap makna semantik dan sintaktik antar kata dengan lebih baik karena dilatih terlebih dahulu pada korpora yang besar (Li dan Yang, 2018).

Implementasi *pre-trained word embedding* pada model DL dapat mempengaruhi akurasi model pembelajaran secara signifikan karena kemampuannya dalam merepresentasikan vektor kata-kata *low-dimensional* (Rabut dkk., 2019; Rios dan Lwowski, 2020). Lebih lanjut, pendekatan ini juga merupakan salah satu bentuk *transfer learning* (Wang dkk., 2020; Torregrossa dkk., 2021; Shahi dkk., 2022). *Transfer learning* merupakan metode *transfer* pengetahuan yang mempelajari satu atau lebih tugas yang telah dilatih sebelumnya dan kemudian digunakan untuk meningkatkan pembelajaran dalam tugas lain yang berkaitan (Olivas dkk., 2009).

Menurut Sanjanasri dkk. (2021) dan Mohamed dkk. (2020) *pre-trained word embedding* yang paling akurat dan umum digunakan untuk klasifikasi teks adalah Word2Vec yang dibangun oleh Mikolov dkk. (2013b), GloVe oleh Pennington dkk. (2014), dan FastText oleh Bojanowski dkk. (2017). Bahkan dalam beberapa tahun terakhir, aplikasi Word2Vec dan Glove dalam klasifikasi teks dari domain biomedis telah banyak ditemukan karena memiliki kemampuan representasi kata yang baik

(Zhao dkk., 2021). Misalnya, penelitian oleh Almazaydeh dkk. (2023) yang menggunakan Word2Vec dalam klasifikasi rekam medis elektronik, dan penelitian oleh Lu dkk. (2022) yang menggunakan GloVe dalam klasifikasi catatan medis. Akan tetapi, Word2Vec, GloVe dan FastText dibangun dengan korpora berita atau Wikipedia dari domain umum. Sementara, AlRashdi dan O'Keefe (2019) menyatakan bahwa kesesuaian domain dan korpora dapat berdampak positif dalam meningkatkan kualitas *word embedding*. Berdasarkan opini tersebut, penggunaan Word2Vec, GloVe dan FastText sebenarnya kurang sesuai untuk domain biomedis.

Saat ini, *pre-trained word embedding* yang tersedia terutama di bidang biomedis hanya berfokus pada penggunaan sumber tunggal korpora teks besar seperti PubMed atau PubMed *Central* (PMC). Beberapa penelitian (Faruqui dkk., 2015; Han dkk., 2016; Yamada dkk., 2016; Cao dkk., 2017) telah menyarankan untuk mengintegrasikan pengetahuan domain dengan korpora teks guna meningkatkan kualitas *word embedding*. Dalam domain biomedis, ada banyak data pengetahuan biomedis seperti judul subjek medis (MeSH) dan sistem bahasa medis terpadu (UMLS), yang dapat dieksplorasi untuk melengkapi informasi tekstual dalam literatur. Secara intuitif, mengintegrasikan pengetahuan domain biomedis semacam itu akan membantu meningkatkan kualitas *word embedding* sedemikian rupa sehingga lebih baik dalam menangkap semantik istilah dan konsep khusus. Oleh karena itu, Zhang dkk. (2019) mengembangkan *pre-trained word embedding* baru untuk domain biomedis bernama BioWordVec yang dibangun dari dua sumber data berbeda, berupa literatur biomedis (PubMed) dan domain pengetahuan (MeSH).

Aplikasi *pre-trained word embedding* dengan konsep NLP pada model DL telah banyak digunakan untuk pengklasifikasian teks, seperti pada model *Convolutional Neural Network* (CNN) (Bai dkk., 2017), dan *Long Short-Term Memory* (LSTM) yang dikembangkan dari arsitektur *Recurrent Neural Network* (RNN). *Convolutional Neural Network* (CNN) merupakan algoritma DL populer dalam menyelesaikan tugas klasifikasi teks pada banyak penelitian. Arsitektur CNN terinspirasi oleh neuron pada otak manusia yang mirip dengan jaringan syaraf konvensional. Aplikasinya tidak terbatas pada klasifikasi teks, tapi dapat ditemukan

secara luas di berbagai bidang, termasuk *computer vision*, *speech processing*, *face recognition*, dan sebagainya karena CNN memiliki kemampuan generalisasi yang baik dalam mempelajari fitur abstrak pada data, terutama data spasial (Alzubaidi dkk., 2021).

Sementara itu, model LSTM merupakan merupakan jenis RNN yang efektif untuk menangani data sekuensial. Model LSTM mampu memproses masukkan panjang variabel serta dapat mengatasi masalah *long-term dependencies*, *exploding* dan *vanishing gradient* sehingga dianggap lebih sesuai untuk klasifikasi teks dibanding CNN (Yan dkk., 2018; Wang dan Li, 2022). Model *hybrid* juga mulai dikembangkan dengan mengkombinasikan keunggulan model CNN dan model LSTM dalam menangkap fitur lokal dan global dari kata dan kalimat secara efektif (Prabhakar dkk., 2021).

Berdasarkan pemaparan di atas, salah satu pendekatan yang dapat digunakan untuk meningkatkan performa model STS dan klasifikasi adalah penggunaan *word embedding*. Penelitian ini memilih menggunakan *pre-trained word embedding* yang dilatih pada korpora yang sesuai dengan domain biomedis bernama BioWordVec, karena *embedding* yang diperoleh melalui data pelatihan untuk tugas tertentu (*custom-trained embedding*) tidak dapat diterapkan pada tugas lainnya. BioWordVec merupakan *pre-trained word embedding* yang dibangun oleh Zhang dkk. (2019) dengan menggabungkan informasi *sub-word* dari teks biomedis dan istilah biomedis pada PubMed dan MeSH. Implementasi BioWordVec sangat sesuai untuk menangani tugas ekstraksi informasi pada data teks biomedis, khususnya GENIA *Biomedical Event* yang berkaitan dengan faktor transkripsi pada sel darah manusia dan leukemia, karena berasal dari sumber yang sama, yaitu PubMed dan MeSH. Sehingga perlu dilakukan kajian untuk mengetahui apakah implementasi BioWordVec pada data GENIA *Biomedical Event* dapat meningkatkan performa model DL untuk ekstraksi informasi.

Penelitian ini mengusulkan implementasi arsitektur *Siamese Manhattan* pada model CNN dan LSTM untuk menentukan STS Biomedis pada kumpulan data

GENIA *Biomedical Event* dengan pendekatan berbasis korpora menggunakan korpora spesifik domain biomedis yang disebut BioWordVec. Kumpulan data GENIA *Biomedical Event* merupakan kumpulan teks biomedis asli yang tidak memiliki label, sehingga model dibangun dengan dua proses pelabelan berbeda menggunakan metode *Cosine Similarity* (CS) dan *Word Mover's Distance* (WMD). Selain itu, penelitian ini juga mengusulkan penggabungan model CNN – LSTM dan LSTM – CNN dengan menerapkan arsitektur *Siamese Manhattan* untuk memodelkan STS pada teks biomedis. Menurut beberapa penelitian dari Beniwal dkk. (2022), Liu dkk. (2018), dan Mansoor dkk. (2020), kombinasi ini memungkinkan model memanfaatkan keunggulan arsitektur CNN dan LSTM untuk memperoleh hasil yang lebih akurat dan tepat dalam menghitung STS.

Sedangkan pada tugas klasifikasi teks, penelitian ini mengusulkan implementasi model DL, antara lain CNN, LSTM, *hybrid* CNN - LSTM, dan *hybrid* LSTM - CNN untuk mengklasifikasikan GENIA *Biomedical Event* berdasarkan kata pemicu atau elemen kunci. Performa model klasifikasi ditingkatkan dengan menangani data tidak seimbang menggunakan kombinasi *undersampling* dan *oversampling*, melakukan *hyperparameter tuning* untuk memperoleh parameter terbaik, dan menerapkan *pre-trained word embedding* khusus domain biomedis, yaitu BioWordVec sebagai fitur masukkan model DL berdasarkan pembagian data menggunakan metode *train-test split* dan *k-fold Cross Validation*.

Hasil penelitian ini diharapkan dapat membantu memahami hubungan semantik antar variabel atau kalimat biomedis pada *biomedical* big data *report* leukemia. Di samping itu, penelitian ini diharapkan dapat membantu peneliti dan profesional medis untuk mengekstraksi informasi penting dan elemen-elemen kunci dari teks biomedis secara otomatis.

1.2 Identifikasi Masalah

1. Proses ekstraksi informasi secara manual membutuhkan lebih banyak *effort*, waktu yang lama, dan kekurangan ekspert pada bidang biomedis yang

signifikan. Oleh karena itu, diperlukan metode alternatif yang lebih efektif dan efisien dalam menganalisis dan mengekstraksi teks biomedis untuk mengatasi pertumbuhan publikasi yang terus meningkat.

2. Proses anotasi atau pelabelan di bidang biomedis memerlukan keahlian profesional medis. Akibatnya, ketersediaan kumpulan data berlabel dalam domain biomedis terbatas dan penelitian terkait STS biomedis terhambat. Oleh karena itu, dibutuhkan metode lain untuk melakukan pelabelan data biomedis.
3. Masalah yang sering muncul dalam tugas klasifikasi teks adalah ketidakseimbangan data. Sementara, adanya ketidakseimbangan data dapat menimbulkan bias dan mempengaruhi performa model klasifikasi. Sehingga, masalah ketidakseimbangan data harus diatasi untuk memperoleh performa model klasifikasi yang baik.
4. Prosedur pembagian data dalam model *deep learning* biasanya menggunakan metode *train-test split*. Namun, metode ini dapat mengakibatkan terjadinya bias dan *overfitting*. Oleh sebab itu, untuk memastikan hasil penelitian tidak mengalami hal tersebut, diterapkan metode k-*fold Cross Validation* dengan cara melipat data menjadi *n-fold* yang memungkinkan setiap bagian data akan menjadi data pelatihan sekaligus data pengujian.
5. Pendekatan yang biasa digunakan untuk meningkatkan performa model STS dan model klasifikasi teks adalah aplikasi *pre-trained word embedding*. Sayangnya, *pre-trained word embedding* yang umum tersedia berasal dari domain umum. Sementara, untuk menganalisis *biomedical* big data *report* yang berasal dari domain biomedis, memerlukan *pre-trained word embedding* dari domain yang sama. Oleh karena itu, untuk meningkatkan performa model STS dan model klasifikasi teks, penelitian ini menerapkan BioWordVec yang merupakan *pre-trained word embedding* khusus domain biomedis.
6. Model tunggal CNN memiliki kemampuan generalisasi yang baik dalam mempelajari fitur abstrak pada data, terutama data spasial. Sedangkan model tunggal LSTM efektif untuk menangani data sekuensial. Sehingga, integrasi kedua model tersebut dapat memanfaatkan keunggulan arsitektur jaringan CNN sekaligus LSTM, serta hasil yang diperoleh lebih tepat dan akurat.

7. Metode yang menjadi pilihan terbaik untuk tugas STS adalah *Siamese Manhattan.* Cara kerja model *Siamese Manhattan* adalah dengan mengkonfigurasikan jaringan sesuai dengan tugas, termasuk penggunaan model DL seperti CNN, LSTM, *hybrid* CNN - LSTM, dan *hybrid* LSTM - CNN. Oleh karena itu, penelitian ini mengaplikasikan arsitektur *Siamese Manhattan* pada model CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN.

1.3 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana cara menerapkan lapisan *embedding* BioWordVec pada ekstraksi informasi, yaitu STS dan klasifikasi teks untuk *biomedical* big data *report*?
2. Bagaimana proses pelabelan data untuk tugas STS apabila data tidak memiliki label?
3. Bagaimana cara menemukan model terbaik dengan mengaplikasikan arsitektur *Siamese Manhattan* pada model *deep learning*, yaitu CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN ?
4. Bagaimana cara mengatasi ketidakseimbangan kelas pada data *biomedical* big data *report* untuk tugas klasifikasi teks?
5. Bagaimana cara membangun dan menemukan model klasifikasi teks terbaik dengan mengaplikasikan prosedur pembagian data berdasarkan metode *train-test split* dan *k-fold Cross Validation* pada model *deep learning*, yaitu CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN ?
6. Bagaimana cara mengevaluasi, menganalisis, dan melakukan perbandingan performa model STS dan model klasifikasi teks?

1.4 Tujuan Penelitian

Berdasarkan rumusan permasalahan yang telah diuraikan pada latar belakang, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Melakukan penerapan lapisan *embedding* BioWordVec pada tugas STS dan tugas klasifikasi teks untuk *biomedical* big data *report.*
2. Melakukan prosedur pelabelan data menggunakan metode *Cosine Similarity* dan *Word Mover's Distance*.
3. Membangun dan menguji model STS terbaik dengan mengaplikasikan arsitektur *Siamese Manhattan* pada model *deep learning*, yaitu CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN.
4. Melakukan *resampling* dengan mengkombinasikan metode *Random Undersampling* dan *Random Oversampling*.
5. Membangun dan menguji model klasifikasi teks dengan mengaplikasikan prosedur pembagian data berdasarkan metode *train-test split* dan *k-fold Cross Validation* pada model DL, yaitu CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN.
6. Melakukan evaluasi, analisis, dan perbandingan performa model STS dan model klasifikasi teks berdasarkan metrik evaluasi, yaitu Akurasi, Presisi, *Recall*, dan *F1-score*.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini antara lain:

1. Meningkatkan performa model dalam melakukan ekstraksi informasi dari data *biomedical* big data *report* melalui aplikasi *pre-trained word embedding* khusus domain biomedis, yaitu BioWordVec.
2. Memberikan sumbangan terhadap pengembangan *data science*, khususnya dalam analisis Big Data untuk mengekstraksi informasi *biomedical* big data *report* tentang penyakit leukemia atau kanker darah dengan HPC.
3. Memberikan sumbangan terhadap bidang biomedis, khususnya sebagai metode alternatif dalam membantu dokter dan praktisi medis untuk memahami *biomedical* big data *report* tentang penyakit leukemia atau kanker darah.

1.6. Batasan Penelitian

Batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Kumpulan Data yang digunakan merupakan kumpulan data publik, yaitu GENIA *Biomedical Event* dalam Bahasa Inggris yang diperoleh dari https://kaggle.com/datasets/nishanthsalian/genia-biomedical-event-dataset
2. BioWordVec digunakan dan dievaluasi dalam dua jenis tugas NLP (BioNLP): intrinsik dan ekstrinsik. Dua jenis kumpulan data BioNLP dibuat dalam format biner berisi 2,324,849 kata yang berbeda secara keseluruhan, terdiri dari 2,309,172 kata berasal dari PubMed dan 15,677 dari MeSH. Setiap kata dikonversi menjadi huruf kecil dan jumlah dimensinya adalah 200.
3. Jumlah label dalam setiap kelas yang digunakan pada tugas klasifikasi teks adalah label pertama pada setiap kelas.

1.7. Keterbaruan (*Novelty*)

Keterbaruan yang dicapai dalam disertasi ini berupa:

1. Kontribusi dalam aplikasi *pre-trained word embedding* khusus domain biomedis untuk meningkatkan performa model dalam melakukan ekstraksi informasi dari data *biomedical* big data *report*. Hasil penelitian menunjukkan bahwa model dapat mengekstraksi informasi penting dan elemen-elemen kunci dari teks biomedis secara otomatis dengan baik.
2. Kontribusi dalam melakukan proses pelabelan pada data tidak berlabel untuk tugas STS *biomedical* big data *report.*
3. Kontribusi dalam menangani masalah ketidakseimbangan data menggunakan kombinasi metode *Random Undersampling* dan *Random Oversampling* pada klasifikasi data *biomedical* big data *report.*
4. Kontribusi dalam memastikan hasil klasifikasi menggunakan model DL tidak terjadi bias dan *overfitting* dengan melakukan pembagian data berdasarkan k-*fold Cross Validation*. Hal ini dikarenakan, penerapan *k-fold Cross Validation* umumnya ditemukan pada metode ML tradisional. Sedangkan pada metode DL, belum ditemukan penelitian serupa.

5. Kontribusi dalam menemukan model terbaik dalam menentukan kesamaan semantik dan melakukan klasifikasi pada *biomedical* big data *report* terkait penyakit leukemia, yaitu model *hybrid* LSTM-CNN.

BAB 2
KAJIAN PUSTAKA

2.1 Penelitian Terkait STS

Beberapa penelitian terkait STS yang telah dilakukan dirangkum pada Tabel 1 sebagai bahan acuan dan perbandingan untuk hasil analisis dalam mengukur kesamaan semantik berdasarkan model, proses pelabelan untuk data yang tidak memiliki label, dan *word embedding* yang digunakan untuk menangkap informasi semantik.

Tabel 1. Penelitian terkait Model *Siamese* untuk STS

Judul	Penulis	Metode	Performa
Unlabeled Short Text Similarity with LSTM Encoder	Yao dkk. (2019)	a. Data: MSR Paraphrase dan Quora Dataset (Tidak memiliki label) b. Pelabelan: Cosine Similarity c. Domain: Umum d. Word embedding: Word2Vec e. Model: LSTM encoder	Akurasi MSR: 88,.00% Recall MSR: 87,00% Akurasi Quora: 85,00% Recall Quora: 92,00%
Semantic Textual Similarity with Siamese Neural Networks	Ranasinghe dkk. (2019)	a. Data: SemEval 2017 Dataset (Memiliki label) b. Domain: Biomedis c. Word embedding: Custom-trained embedding d. Model: Siamese Manhattan – LSTM	Akurasi: 86,51%
Detection of Medical Text Semantic Similarity Based on Convolutional Neural Network	Zheng dkk. (2019)	a. Data: Imaging & Pathology Report-pairs (Memiliki label) b. Domain: Biomedis c. Word embedding: CMESH d. Model: Siamese Neural Network – CNN dengan algoritma LIME untuk menentukan kesamaan	Recall: 93,70% Presisi: 94,50% F1-Score: 94,10%

Judul	Penulis	Metode	Performa
A Siamese CNN Architecture for Learning Chinese Sentence Similarity	Shi dkk. (2020)	a. Data: Chinese Sentence Pairs (Memiliki label) b. Domain: Umum c. Word embedding: Custom-trained embedding d. Model: Siamese Neural Network – CNN dengan membandingkan dua metrik jarak (Cosine & Manhattan)	Akurasi Cosine: 77,05% Akurasi Manhattan: 77,31%
Siamese KG – LSTM: A deep learning model for enriching UMLS Metathesaurus synonymy	Tran dkk. (2020)	a. Data: The Unified Medical Language System Metathesaurus Dataset (Memiliki label) b. Domain: Biomedis c. Word embedding: BioWordVec d. Model: Siamese Manhattan - LSTM dengan Knowledge Graph	Akurasi: 98,23% Recall: 93,87% Presisi: 97,86% F1-Score: 96,41%
Similarity Matching of Medical Question based on Siamese Network	Li and He (2023)	a. Data: Ethnic Medical Question Dataset (Memiliki label) b. Domain: Biomedis c. Word embedding: Word2Vec Model: Siamese Neural Network – RNN dengan Jarak Manhattan	F1-Score Euclidean: 94,13% F1-Score Cosine: 96,93% F1-Score Manhattan: 97,34%

Yao dkk. (2019) mengusulkan algoritma baru untuk menghitung kesamaan teks pendek yang tidak memiliki label menggunakan model berbasis LSTM *encoder*. Pada tahap *preprocessing*, teks terlebih dahulu diberi label dengan cara menghitung kesamaan berdasarkan jarak *Cosine* yang disebut *Cosine Similarity*. Kemudian, setiap kata dalam teks direpresentasikan menggunakan Word2Vec. Proses pelatihan memanfaatkan lapisan *embedding* untuk mengekstrak lebih banyak fitur dari berbagai dimensi dan meningkatkan sel LSTM dalam mempelajari informasi urutan kata dari teks pendek serta melupakan informasi yang dikumpulkan sebelumnya. Penelitian dilakukan pada dua kumpulan data, yaitu kumpulan data *Microsoft Research* (MSR) *Paraphrase Corpus* dan Quora QR. Model dibandingkan dengan model lain, yaitu TF – IDF, dan *edit distance*. Hasil penelitian menunjukkan bahwa kesamaan teks pendek menggunakan model LSTM *encoder* dan *Cosine Similarity*

sebagai metode pelabelan memiliki akurasi dan *recall* yang lebih tinggi daripada model lainnya pada dua kumpulan data yang berbeda dengan akurasi mencapai 88% untuk data MSR *Paraphrase Corpus* dan 85,00% untuk data Quora QR. Sehingga, data yang tidak memiliki label dalam tugas STS dapat ditangani menggunakan *Cosine Similarity* sebagai metode pelabelan.

Ranasinghe dkk. (2019) mengukur kesamaan semantik pada teks menggunakan *Siamese Neural Network* – LSTM dimana masukkannya direpresentasikan dengan vektor kata yang telah dilatih secara terpisah pada korpora Wikipedia yang sering disebut sebagai *pre-trained word embedding* Word2Vec. Teks yang diukur kesamaan semantik-nya adalah pasangan kalimat pada data SemEval 2017 yang memiliki 8,277 pasangan kalimat berlabel. Arsitektur *Siamese Neural Network* yang digunakan terdiri dari lapisan *embedding* yang merepresentasikan setiap kalimat sebagai rangkaian vektor kata. Urutan vektor kata ini dimasukkan ke dalam sel LSTM yang mempelajari pemetaan berdasarkan urutan panjang variabel vektor 300 dimensi ke dalam vektor 50 dimensi. Selanjutnya, jarak *Manhattan* digunakan untuk menentukan kesamaan pada lapisan *output*. Hasil penelitian menunjukkan bahwa model *Siamese Neural Network* – LSTM menggunakan jarak *Manhattan* memiliki performa yang cukup baik dengan akurasi 86,51%. Perlu diketahui, mereka juga bereksperimen dengan metrik jarak dan *word embedding* lainnya. Namun, penggunaan jarak *Euclidean* sebagai pengganti jarak *Manhattan* hasilnya tidak lebih baik dalam mengukur kesamaan semantik teks. Selain itu, mengubah *word embedding* menjadi GloVe, FastText, atau menggabungkannya dengan model Word2Vec juga tidak mendapatkan hasil yang lebih baik. Oleh karena itu, mereka memilih untuk tidak menampilkan hasil eksperimen tersebut pada jurnal hasil penelitian.

Zheng dkk. (2019) mengusulkan *Siamese Neural Network* berbasis CNN atau dikenal dengan nama *Siamese Neural Network* – CNN untuk mempercepat proses pengambilan informasi. Data masukkan berupa 16,354 pasangan laporan pencitraan dan patologi dari 1,926 pasien yang dirawat di Rumah Sakit Shanghai Tongren dan menjalani pemeriksaan ultrasonik antara 1 Mei 2017 dan 31 Juli 2017. Setiap

pasangan berisi dua laporan, satu laporan pencitraan dan yang lainnya adalah laporan patologi. Model CNN diadaptasi untuk menghitung kesamaan di antara pasangan laporan tersebut untuk mengidentifikasi pasangan laporan target dengan *body sites* yang tumpang tindih, dan membandingkan kinerjanya dengan enam model konvensional lainnya, termasuk *keyword mapping*, LSA, LDA, Doc2Vec, dan *Siamese* – LSTM. Secara umum, arsitektur model yang diusulkan terdiri dari lapisan *input*, lapisan *feature extraction* atau *embedding*, dan lapisan *fully connected*. Lapisan *input* memetakan setiap kata menjadi vektor *dense* dengan 128 dimensi dan mengubah setiap laporan menjadi matriks *dense*. Setiap vektor *dense* mewakili informasi semantik dari kata terkait, yang nilainya dapat diperbarui selama pelatihan. Untuk menginisialisasi vektor kata, mereka menggunakan *randomly initialized embedding* dan *pre-trained word embedding* yang diperoleh dari *Baidu Encyclopedia Corpus* dari domain umum. Selain itu, mereka juga menggunakan metode *graph embedding* sebagai strategi inisialisasi vektor kata ketiga untuk meningkatkan representasi kata dengan menangkap informasi hubungan semantik dari ontologi medis. *Graph embedding* yang digunakan adalah *Chinese Medical Subject Headings* (CMeSH), atau MeSH versi China, yang berisi sekitar 391,892 konsep medis dan 2,047,749 relasi. Selanjutnya, untuk mengidentifikasi fitur atau kata mana yang menentukan hasil prediksi dan meningkatkan kemampuan interpretasi model digunakan algoritma *Local Interpretable Model-Agnostic Explanations* (LIME). Hasil eksperimen menunjukkan bahwa model *Siamese – CNN* yang menggunakan *embedding* sebagai metode representasi kata memperoleh peningkatan yang signifikan dibandingkan dengan semua model konvensional lainnya baik dari presisi, *recall*, maupun *f1-score* yang nilainya melampaui 90% dalam kumpulan data pengujian. Namun, model *Siamese* – CNN yang menggunakan *graph embedding* dengan domain biomedis terbukti memiliki performa yang lebih baik dibandingkan *randomly initialized embedding* dan *pre-trained word embedding* dari domain umum dengan presisi sebesar 93,70%, *recall* sebesar 94,50%, dan *f1-score* sebesar 94,10%. Sehingga, dapat disimpulkan bahwa penggunaan *embedding* yang sesuai dengan domain penelitian pada tugas STS dapat meningkatkan performa model dalam

mengukur kesamaan semantik dan berpotensi mempercepat proses pengambilan informasi.

Shi dkk. (2020) mengusulkan model *Siamese Neural Network* – CNN untuk mempelajari kesamaan semantik antara dua pasangan kalimat bahasa Mandarin pada kumpulan data bernama *Large-scale Chinese Question Matching Corpus* (LCQMC) dengan distribusi data yang seimbang, yaitu pasangan kalimat serupa dan pasangan kalimat berbeda masing-masing menempati 50% dan 50% dari seluruh kumpulan data yang diperoleh dari korpus Baidu. Kumpulan data tersebut terdiri dari 283,000 data yang dibagi menjadi 250,000 data sebagai data pelatihan, dan 12,500 data sebagai data pengujian. Arsitektur model dibangun dengan menggunakan dua lapisan konvolusi, di mana masukkan dari setiap lapisan konvolusi adalah *character-level embeddings*, dan keluaran dari setiap lapisan konvolusi adalah representasi tingkat kalimat. Kemudian, jarak *Cosine* dan *Manhattan* digunakan sebagai metrik kesamaan untuk membandingkan keluaran dari dua lapisan konvolusi. Hasil penelitian menunjukkan bahwa jarak *Manhattan* lebih unggul dalam menghitung kesamaan antara dua kalimat dengan akurasi 77,31%, sedangkan jarak *Cosine* 77,05%. Oleh karena itu, dapat disimpulkan bahwa jarak *Manhattan* dapat membantu untuk mencapai konvergensi yang lebih cepat dan akurasi yang lebih tinggi daripada metrik kesamaan lainnya.

Tran dkk. (2020) memanfaatkan model DL LSTM yang mengadopsi arsitektur *Siamese Neural Network* dengan menambahkan fitur semantik yang diekstraksi dari pengetahuan *graph*, bernama *Siamese* KG – LSTM untuk memprediksi sinonim dan bukan sinonim antara pasangan istilah biomedis dalam *The Unified Medical Language System* (UMLS). *The Unified Medical Language System* adalah gudang terminologi medis yang dikembangkan oleh Perpustakaan Kedokteran Nasional AS untuk meningkatkan kemampuan sistem komputer dalam memahami bahasa biomedis dan kesehatan. *Metathesaurus* UMLS merupakan salah satu dari tiga sumber pengetahuan UMLS yang memuat istilah-istilah kedokteran dan hubungannya. Model pembelajaran yang diusulkan berfokus pada sebagian istilah tertentu, bukan keseluruhan korpora *Metathesaurus*. Bobot awal pada model

dihasilkan dari BioWordVec yang telah dilatih sebelumnya pada kumpulan kosakata biomedis PubMed dan MeSH. Model *sub-word embedding* pada BioWordVec memungkinkan model pembelajaran untuk mempelajari kosakata langka pada *Metathesaurus*. Vektor leksikal, yang merupakan keluaran dari lapisan LSTM, akan digabungkan dengan vektor konteks terkait yang diekstraksi dari pengetahuan *graph* sebagai strategi pengayaan kosa kata yang berbeda. Vektor gabungan kemudian memasuki dua lapisan *dense* untuk membuat vektor gabungan. Keluaran model adalah skor kesamaan antara dua vektor gabungan, yang diukur dengan jarak *Manhattan*. Jika nilainya lebih besar atau sama dengan 0,5, maka pasangan istilah biomedis yang di masukkan adalah sinonim; jika tidak, pasangan istilah biomedis tersebut adalah bukan sinonim. Kinerja model dievaluasi berdasarkan klasifikasi sinonim dan bukan sinonim saat menambahkan istilah baru ke dalam kumpulan data dengan empat kali eksperimen. Hasil penelitian menunjukkan bahwa model *Siamese* – KG – LSTM yang menerapkan BioWordVec sebagai *word embedding* untuk memprediksi sinonim dan bukan sinonim pada istilah kata biomedis memiliki performa yang sangat baik dengan akurasi 98,23%, *recall* 98,37%, presisi 97,40%, dan *f1-score* 96,41%

Li dan He (2023) membangun model RNN menggunakan arsitektur *Siamese Neural Network* dan menerapkan Word2Vec untuk pemrosesan vektor kata pada kumpulan data pertanyaan *ethnic medical* yang berisi total 22,655 pasangan pertanyaan. Setiap pasang pertanyaan terdiri dari dua pertanyaan dan sebuah label, dengan label ditetapkan ke 1 jika kedua pertanyaan tersebut memiliki makna semantik yang sama dan 0 jika yang terjadi adalah kebalikannya, dengan separuh dari pasangan pertanyaan masing-masing memiliki label 0 dan 1. Selama pelatihan, kumpulan data dibagi menjadi data pelatihan dan data pengujian berdasarkan skema 80% : 20%, dengan rincian jumlah data 18,124 pada data pelatihan dan jumlah data 4,531 pada data pengujian. Keluaran vektor setelah diproses oleh *Siamese Neural Network* akan dihitung dengan jarak *Euclidean*, jarak *Cosine*, dan jarak *Manhattan* untuk memastikan apakah pasangan pertanyaan tersebut serupa secara semantik. Menurut hasil penelitian, jarak *Manhattan* menghasilkan kesamaan terbaik dengan *f1-score* mencapai 97,34%, sementara jarak Euclidean 94,13% dan jarak *Cosine* 96,93%.

2.2 Penelitian terkait Klasifikasi Teks

Penelitian terkait klasifikasi teks menggunakan model *hybrid* dirangkum pada Tabel 2 dan menjadi bahan acuan dan perbandingan untuk hasil analisis berdasarkan performa model. *Word embedding*, metode *resampling* (apabila data tidak seimbang), dan model *hybrid* yang digunakan adalah topik penelitian yang menjadi perbandingan.

Tabel 2. Penelitian terkait Model *Hybrid* untuk Klasifikasi Teks

Judul	Penulis	Metode	Performa
LSTM – CNN Hybrid Model for Text Classification	Zhang dkk. (2018)	a. Data: Movie Review Dataset (Data seimbang) b. Domain: Umum c. Word Embedding: Custom-trained Embedding d. Model: Hybrid LSTM – CNN e. Klasifikasi: Binary (2 kelas)	Akurasi Validasi: 87,31% Uji: 91,17%
A Hybrid CNN – LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis	Rehman dkk. (2019)	a. Data: IMDB Dataset (Data seimbang) b. Domain: Umum c. Word Embedding: Word2Vec d. Model: Hybrid CNN – LSTM	Akurasi: 91% Recall: 83% Presisi: 91% F1-Score: 87%
SSMFN: a fused spatial and sequential deep learning model for methylation site prediction	Lumbanraja dkk. (2021)	a. Data: Methylation Site from Uniprot Protein Database (Data tidak seimbang) b. Domain: Biomedis c. Word Embedding: Custom-trained Embedding d. Model: Hybrid CNN – LSTM untuk Data seimbang dan Data tidak seimbang e. Klasifikasi : Binary (2 kelas)	Akurasi: 81,15% (seimbang) 84,62 (tidak seimbang) F1-Score: 81,15% (seimbang) 84,35 (tidak seimbang) Sensitivity dan Specificity : 80,00% dan 82.44% (seimbang) 96,88 dan 77,44% (tidak seimbang)
Medical – Based Text Classification Using FastText Features and CNN – LSTM Model	Zeghdaoui dkk. (2021)	a. Data: Medical Reports (Data seimbang) b. Domain: Biomedis c. Word Embedding: FastText d. Model: Hybrid CNN – LSTM f. Klasifikasi : Multikelas	Akurasi: 86,34% Recall: 90,68% Presisi: 91,72% F1-Score: 90,67%

Judul	Penulis	Metode	Performa
MCNN – LSTM: Combining CNN and LSTM to Classify MultiClass Text in Imbalanced News Data	Hasib dkk. (2023)	a. Data: HuffPost News Dataset (Data tidak seimbang) b. Domain: General c. Resampling: Under Sampling d. Word Embedding: Custom-trained Embedding e. Model: Hybrid CNN – LSTM e. Klasifikasi: Multikelas	Akurasi: 99,71% F1-Score: 98,00%

Penelitian oleh Zhang dkk. (2018) membandingkan model CNN, LSTM, dan *hybrid* LSTM – CNN untuk klasifikasi teks berdasarkan objektivitas dan subjektivitas pada data ulasan film. Total data yang digunakan adalah 10.000 ulasan film yang mencakup 5.000 data diperoleh dari *Internet Movie Database* (IMDB) dan 5.000 data lainnya diperoleh dari *Rotten Tomatoes Website*. Data tersebut dibagi menjadi data pelatihan, data validasi, dan data pengujian dengan persentase 81%, yaitu 8.100 data pelatihan, 9%, yaitu 900 data validasi, dan 10%, yaitu 1.000 data pengujian. Kata-kata pada teks direpresentasikan menggunakan *custom-word embedding* yang diperoleh dari data pelatihan. Hasil penelitian menunjukkan bahwa kinerja model CNN cukup baik dengan akurasi yang diperoleh sebesar 80,32% pada data validasi dan 84,17% pada data pengujian. Namun apabila dibandingkan model CNN, kinerja model LSTM pada klasifikasi review film lebih baik dibandingkan model CNN dengan akurasi sebesar 83,45% pada data validasi dan 88,03% pada data pengujian. Sedangkan, tingkat akurasi model *hybrid* LSTM – CNN pada data pengujian dan data validasi adalah 87,31% dan 91,17%, yaitu 6,99% dan 7,00% lebih tinggi dibandingkan model CNN, serta 3,86% dan 3,14% lebih tinggi dibandingkan model LSTM. Oleh karena itu, model *hybrid* LSTM – CNN lebih unggul dibandingkan model CNN maupun LSTM dalam klasifikasi ulasan film.

Rehman dkk. (2019) mengusulkan model *hybrid* CNN – LSTM untuk melakukan analisis sentimen terhadap ulasan film dengan menerapkan Word2Vec sebagai *word embedding*. Model tersebut dibandingkan dengan model DL lain, yaitu CNN dan LSTM. Data yang digunakan adalah data IMDB berisi 40,000 ulasan film dengan dua label dan dibagi dengan rasio 80% untuk data pelatihan, dan 20% untuk

data pengujian. Distribusi label pada setiap data adalah seimbang. Untuk meningkatkan akurasi model, mereka menerapkan *dropout*, dan normalisasi. Hasil dari penelitian menunjukkan bahwa model *hybrid* CNN – LSTM melampaui akurasi, *recall*, dan *f1-score* dari model CNN dan LSTM dengan akurasi mencapai 91%, *recall* sebesar 86%, dan *f1-score* sebesar 89%. Nilai akurasi, *recall*, dan *f1-score* yang diperoleh model CNN adalah 87%, 83%, dan 84%. Sedangkan akurasi, *recall*, dan *f1-score* yang diperoleh untuk model LSTM, yaitu 88%, 83%, dan 86%. Dengan demikian, dapat disimpulkan bahwa model *hybrid* lebih akurat dibandingkan model tunggal.

Lumbanraja dkk. (2021) membangun model gabungan CNN dan LSTM untuk memprediksi kemungkinan situs metilasi pada urutan protein menggunakan data yang diperoleh dari penelitian sebelumnya oleh Kumar dkk. (2017) dan *database* protein Uniprot. Kumpulan data tersebut dibagi menjadi tiga bagian, yaitu data pelatihan, data validasi, dan data pengujian. Setiap kumpulan data berisi sampel positif dan negatif. Karena kumpulan data asli tidak seimbang, maka model *hybrid* CNN – LSTM akan dibandingkan berdasarkan performa model pada data seimbang dan data tidak seimbang. *Convolutional Neural Network* diterapkan untuk mengekstrak informasi spasial, sementara LSTM diterapkan pada data sekuensial. Secara umum, performa model yang diperoleh mencapai kinerja terbaik pada data tidak seimbang dibandingkan data seimbang. Model *hybrid* CNN – LSTM pada data tidak seimbang memperoleh akurasi sebesar 84,62%, *f1-score* sebesar 84,35%, *sensitivity* sebesar 96,88%, dan *specificity* sebesar 77,44%. Sedangkan model *hybrid* CNN – LSTM pada data seimbang memperoleh akurasi sebesar 81,15 %, *f1-score* sebesar 81,15%, *sensitivity* sebesar 80,00%, dan *specificity* sebesar 82,44%.

Zeghdaoui dkk. (2021) mengusulkan model klasifikasi teks biomedis yang mengkombinasikan model CNN dan LSTM untuk laporan medikal yang diperoleh dari berbagai pusat pengendalian kanker di Prancis. Laporan tersebut berisi berbagai informasi medis, seperti keputusan klinis yang dibuat oleh dokter, pengobatan yang diberikan kepada pasien, dan riwayat penyakit yang mencakup, antara lain, operasi sebelumnya, rawat inap, serta penyakit kronis pada anggota

keluarga. Data dibagi menjadi data pelatihan dan data pengujian dengan rasio 80 : 20, di mana 80% data digunakan untuk melatih model dan 20% data digunakan untuk mengevaluasi model. Model tersebut juga menerapkan *pre-trained word embedding* bernama FastText sebagai pendekatan untuk representasi kata. Selanjutnya, performa model dievaluasi dan dibandingkan dengan model lainnya seperti CNN, SVM, *Decision Tree*, *Naïve Bayes*, dan *K-nearest neighbor* menggunakan akurasi, presisi, *recall*, dan *f1-score* sebagai metrik evaluasi. Hasil penelitian menunjukkan bahwa model yang diusulkan mencapai skor akurasi tertinggi, yaitu 86,34% dan mengungguli semua model lainnya dalam semua parameter evaluasi. Secara umum, hasil tersebut mengkonfirmasi bahwa model DL memiliki kinerja yang lebih baik dibandingkan model ML tradisional dalam tugas klasifikasi teks. Selain itu, penggunaan *pre-trained word embedding* FastText sebagai teknik ekstraksi fitur dapat meningkatkan akurasi model.

Terbaru, Hasib dkk. (2023) mengusulkan sebuah model bernama *Multiclass* CNN – LSTM (MCNN – LSTM) yang menggabungkan dua model DL, yaitu CNN dan LSTM untuk klasifikasi teks pada data *HuffPost News*. *HuffPost News* merupakan situs web yang menampilkan berbagai topik seperti berita, blog, dan konten asli. Himpunan data ini berisi berita utama dan deskripsi singkat dari berbagai surat kabar selama 6 tahun dengan total data sekitar 202,372. Data dikelompokkan ke dalam 12 jenis kelas yang berbeda, di antaranya: *Lifestyle & Wellness*, *Politics*, *Sports & Entertainment*, *Travel Tourism & Art Culture*, *Empowered Voices*, *Parenting & Education*, *Misc*, *General*, *Worldnews*, *Business & Money*, *Science & Tech*, dan *Environment.* Jumlah data setiap kelas tidak seimbang dengan *Lifestyle & Wellness* sebagai kelas mayoritas, sebanyak 14,750 dan *Environment* sebagai kelas minoritas, sebanyak 1,450. Metode yang digunakan untuk menangani kelas tidak seimbang adalah algoritma *Tomek-Links* yang merupakan salah satu metode *under sampling*. Algoritma tersebut bertujuan menghilangkan batas antara kelas mayoritas dan minoritas dengan cara menghapus data dari kelas mayoritas. Ketika distribusi data telah seimbang, data dibagi menjadi 70% data pelatihan dan 30% data pengujian. Selama proses pelatihan, setiap kata dikodekan sebagai vektor *low-dimensional* melalui lapisan *embedding* menggunakan *pre-trained word embedding*

Word2Vec. Selain itu, mereka juga menerapkan *dropout* pada model untuk mencegah *overfitting*. Hasil penelitian menunjukkan bahwa model *hybrid* CNN – LSTM yang diterapkan pada data yang sudah seimbang memiliki performa yang lebih baik dengan akurasi mencapai 99,71% dan *f1-score* mencapai 98,00%, dibandingkan model *hybrid* CNN – LSTM pada data tidak seimbang yang hanya memperoleh akurasi sebesar 95,40% dan *f1 score* sebesar 95,00%. Hal ini berarti bahwa performa model pada tugas klasifikasi teks *multiclass* dapat ditingkatkan dengan menangani data tidak seimbang.

2.3 *Natural Language Processing*

Natural Language Processing (NLP) adalah cabang ilmu komputer yang berkembang dari studi bahasa dan komputasi linguistik dikenal dengan nama *computational linguistics* dalam kecerdasan buatan yang membahas tentang interaksi antara manusia dengan komputer menggunakan bahasa alami, yakni bahasa manusia. Bahasa alami yang dimaksud adalah pesan atau pernyataan yang dikomunikasikan oleh manusia baik secara lisan maupun tulisan (Krishnan dan Rogers, 2015; Gudivada dan Arbabifard, 2018)

Pembelajaran dalam NLP merupakan pengembangan teknik yang bertujuan agar komputer dapat memahami bahasa manusia. Bahasa alami yang digunakan oleh manusia dari berbagai negara memiliki perbedaan dalam bentuk penulisan dan pengucapan, sehingga NLP berupaya memecahkan masalah untuk memahami bahasa manusia dengan semua aturan gramatika dan semantiknya. Proses komputasi bahasa harus direpresentasikan sebagai rangkaian simbol yang memenuhi aturan tertentu.

Secara sederhana, NLP mencoba untuk membuat komputer dapat memahami dan mengerti perintah-perintah yang ditulis dalam bahasa standar manusia atau memahami teks dan kata-kata yang diucapkan seperti yang dipahami manusia. Sehingga, NLP dapat didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam

percakapan sehari-hari. Tujuan NLP adalah untuk mengembangkan dan merancang aplikasi yang mampu menjadi penengah antara interaksi manusia dan mesin dengan perangkat lain melalui penggunaan bahasa alami (Pustejovsky dan Stubbs, 2013). Komputer dapat mengenali, merespon komunikasi dan mengerti perintah-perintah yang ditulis dalam bahasa alami menggunakan algoritma komputer dan kecerdasan buatan. Dengan demikian, NLP menangani pemrosesan dan analisis dokumen secara otomatis, dengan sedikit atau tanpa campur tangan manusia agar komputer meniru cara manusia berkomunikasi dan memahaminya.

Menurut Redd (2014), ada dua teknik utama untuk memahami pengolahan bahasa, yaitu:

1. Analisis sintaksis: teknik pengaturan pada suatu kalimat sehingga kalimat memiliki tata bahasa yang benar. Analisis sintaksis melibatkan penentuan struktur seperti subjek, objek, predikat, dan sebagainya. Komputer akan membaca masukkan kalimat yang dipecah menjadi kata per kata, kemudian menghasilkan deskripsi kalimat yang terstruktur. Teknik ini digunakan untuk menyederhanakan suatu kalimat agar memudahkan pencarian informasi dan juga membantu mendeteksi keberadaan kata atau kalimat baru yang tidak biasa. Contohnya kalimat “Nurul makan”, dengan teknik analisis sintaksis, komputer dapat membedakan mana subjek (“Nurul”) dan predikat (“makan”). Kalimat tersebut mungkin saja tidak memiliki makna atau arti tertentu, karena analisis sintaksis hanya memastikan bahwa struktur dari sebuah kalimat sudah benar.
2. Analisis semantik: teknik yang digunakan untuk memahami makna atau interpretasi tertentu dari suatu kalimat yang digunakan untuk kalimat atau struktur bahasa. Manusia bisa memahami suatu perkataan berdasarkan intuisi dan pengetahuan dari bahasa itu sendiri, tetapi komputer tidak. Komputer membutuhkan teknik, yaitu analisis semantik. Semantik adalah proses penting, karena semantik mendeskripsikan makna yang terkandung pada sebuah kalimat atau kata. Semantik juga mempelajari hubungan antara berbagai konsep dalam teks. Contohnya, apabila sebuah kalimat mengandung kata “uang” dan “accounting”, maka topik yang sedang dibahas berkaitan dengan ekonomi.

Untuk mencapai tujuan proses pemahaman bahasa alami dibutuhkan tiga tahapan proses. Proses pertama dan kedua meliputi analisis sintaksis dan semantik, sedangkan proses terakhir adalah interpretasi kontekstual dari keseluruhan kalimat atau segmentasi teks (Lee, 2020). Menurut Levy dkk. (2021), dalam proses NLP, terdapat beberapa kendala dikarenakan perkembangan bahasa alami manusia. Berikut ini adalah sub area NLP:

1. *Phonology/Phonetic*, yaitu sub area yang terkait dengan suara atau ucapan yang diinterpretasikan menjadi kata (*speech-based system*).
2. *Morphology*, yaitu sub area yang membahas struktur dari kata, bentuk kata dasar, dan bentuk kata imbuhan (prefiks dan sufiks).
3. *Syntax*, yaitu sub area yang terkait dengan cara kata-kata digunakan untuk membentuk urutan kata dalam sebuah kalimat.
4. *Semantics*, yaitu sub area yang terkait dengan arti kata dari sebuah kalimat.
5. *Pragmatics*, yaitu sub area yang membahas konteks kata atau kalimat yang memiliki hubungan dengan keadaan.

2.4 *Text Mining*

Text Mining (TM) adalah proses ekstraksi informasi dari data teks dalam jumlah besar dan identifikasi serta eksplorasi pola yang menarik dalam data tekstual (Feldmen dan Sanger, 2007). Proses TM bertujuan untuk mencari kata yang dapat mewakili isi dari data teks tersebut, sehingga dapat dilakukan analisis hubungan antar data teks. Sumber data teks yang digunakan pada TM adalah sekumpulan dokumen dengan format tidak terstruktur. Sehingga, untuk mengubah data teks tersebut menjadi ringkasan yang bermakna, TM harus mempersiapkan kumpulan teks menjadi lebih terstruktur dan selanjutnya melakukan perhitungan bobot frekuensi kemunculan kata.

Text mining telah terbukti sebagai strategi yang unggul untuk identifikasi, normalisasi, dan disambiguasi entitas kunci dalam menyelidiki hubungan di antara entitas (Ananiadou dkk., 2010). Teknologi TM yang ada saat ini telah mencapai hasil yang baik dalam abstraksi otomatis, mesin penjawab otomatis, *relasional*

network, dan resolusi *anaphora* (Lin dkk., 2012; Chen dkk., 2013). Miner dkk. (2012) menyatakan ada tujuh area penerapan TM, yaitu temu kembali informasi, klasifikasi, klasterisasi, *web mining*, ekstraksi informasi, NLP, dan ekstraksi konsep. Menurut Chaubey dan Adebayo (2019), salah satu tugas khusus dalam TM adalah kategorisasi atau klasifikasi data teks.

Kajian analisis data teks berbasis TM pernah dilakukan oleh disiplin ilmu yang bervariasi, antara lain: Neto dkk. (2000) menjelaskan kerangka dasar proses klasterisasi dokumen melalui pendekatan TM; Ochieng dan Djatna (2014) menerapkan TM untuk melakukan klasterisasi berita utama pada situs portal berita dengan algoritma *k-means*; Gurusamy dan Subramaniam (2017) melakukan penelitian mengenai pengelompokan perilaku pengguna media sosial dengan teknik klasterisasi berbasis algoritma *k-means*; dan Allahyari dkk. (2017) melakukan studi mengenai teknik dan aspek fundamental dalam penerapan TM untuk bidang kesehatan dan biomedis.

Text mining merupakan suatu proses ekstraksi informasi dalam jumlah besar dari data teks yang tidak terstruktur untuk dianalisis dan diolah menjadi pemahaman baru yang sebelumnya tidak diketahui sehingga menjadi sebuah pengetahuan (Mangat dan Saini, 2022). Analisis TM memerlukan pendekatan yang berbeda dibandingkan analisis data numerik biasa. Hal khusus yang perlu dilakukan dalam analisis TM adalah teks *preprocessing*.

Teks *preprocessing* adalah suatu proses pembersihan data teks untuk menangani sejumlah besar kata yang tidak terstruktur seperti singkatan, emotikon, simbol, dan angka. Teknik *preprocessing* diperlukan untuk mengekstrak informasi dan mengubah teks tidak terstruktur menjadi bentuk vektor. Crone dkk. (2006) dan Alshdaifat dkk. (2021) menyatakan bahwa *preprocessing* memiliki dampak yang signifikan pada akurasi model dalam TM. Langkah-langkah dalam *preprocessing* data teks meliputi:

1. *Case Folding*

   Case folding merupakan proses mengubah semua karakter huruf kapital menjadi huruf kecil (*lower case*) dalam suatu dokumen. Hal ini perlu dilakukan agar teks menjadi seragam.

2. *Filtering/Cleansing*

   Filtering/cleansing merupakan proses menghilangkan karakter khusus yang tidak diperlukan dalam proses *training*, seperti tanda baca, angka dan simbol lainnya.

3. *Stopword Removal*

   Stopword removal merupakan proses menghilangkan kosa kata yang bukan merupakan ciri (kata unik) dari suatu dokumen dan kata-kata gaul atau slang serta kata-kata yang tidak memiliki makna dalam kamus bahasa yang digunakan, seperti kamus bahasa Inggris atau Indonesia (sastrawi). Contoh dalam Bahasa Inggris seperti *is, are, to, was*, dan sebagainya.

4. *Stemming* atau *Lemmatization*

   Stemming dan *lemmatization* keduanya mereduksi infleksi dan bentuk terkait turunan yang sesuai dari sebuah kata menjadi bentuk dasar yang sama. *Stemming* adalah proses menghilangkan variasi bentuk tata bahasa dari kata yang sama, contohnya, "connection" - "connect", "computing" - "compute", dan lain-lain. Proses *stemming* diselesaikan untuk meningkatkan efektivitas proses teks agar tidak terjadi ketidaksesuaian baik pada prefiks maupun sufiks. Faktanya, hal ini dapat menyebabkan ketidakcocokan atau bahkan kehilangan informasi yang relevan. *Lemmatization* hampir sama dengan s*temming*. *Lemmatization* merupakan proses menghilangkan akhiran infleksi dan mengembalikan suatu kata menjadi kata dasar atau bentuk kamus dengan mengetahui konteks dari kata tersebut. Sebagai contoh, untuk kata “saw”, teknik *lemmatization* mengembalikan “see” atau “saw”, tergantung pada apakah kata tersebut digunakan sebagai kata kerja atau sebagai kata benda.

5. *Tokenization*

   Tokenization merupakan proses memecah kalimat menjadi kata demi kata dengan cara mengurai kalimat tersebut menjadi kosa kata.

6. *Word Tokenization*

   Word tokenization merupakan proses mengubah kata-kata dalam kalimat menjadi nilai indeks dalam bentuk numerik. Proses ini menetapkan nilai indeks berdasarkan kata yang sering muncul dalam kumpulan data.

7. *Padding* Data

   Padding data adalah proses mengisi vektor dengan angka 0 dan *string* kata kosong untuk memastikan kumpulan data memiliki panjang vektor yang sama. Hal ini disebabkan setiap pesan teks dalam kumpulan data memiliki jumlah kata yang berbeda, sehingga perlu dilakukan *padding* data agar panjang pesan teks yang kurang dari panjang maksimal memiliki panjang vektor yang sama.

2.5 *Machine Learning*

Istilah *Machine Learning* (ML) pertama kali dikenalkan oleh seorang ilmuwan komputer bernama Arthur Samuel pada tahun 1959. Pada awalnya, istilah ini muncul dalam konteks permainan catur, di mana komputer dapat belajar dan mengembangkan kemampuan untuk bermain catur dengan sendirinya (Joshi, 2020). Namun, seiring dengan berjalannya waktu, konsep ML berkembang pesat dan mencakup berbagai bidang dalam ilmu komputer dan teknologi. *Machine Learning* merupakan salah satu cabang ilmu statistik terapan yang lebih menekankan pada penggunaan algoritma komputer dengan kapasitas komputasi yang besar untuk mengolah kumpulan data (Goodfellow dkk., 2016; Triguero dan Galar, 2023).

Machine Learning identik dengan kecerdasan buatan, yaitu suatu cabang ilmu yang mempelajari cara mengembangkan sistem komputer agar mampu belajar dari data, membuat algoritma, dan menjalankan tugas-tugas tanpa intervensi manusia (El Naqa dan Murphy, 2015). Ide dasar ML adalah meningkatkan kecerdasan yang mencakup kemampuan individu atau mesin untuk belajar tanpa batasan. Mesin yang dapat belajar akan meningkatkan produktivitas pengguna dan memiliki kemampuan yang unik. Kemampuan belajar adalah domain utama yang ditentukan berdasarkan kemampuan perangkat lunak atau algoritmanya.

Secara mendasar, algoritma ML bekerja dengan cara menciptakan sebuah algoritma yang memungkinkan mesin untuk mempelajari data secara otomatis (Jiao, 2020). *Machine Learning* mengacu pada kemampuan komputer dalam mempelajari data yang ada dan menghasilkan perilaku atau keputusan tanpa perlu diprogram secara eksplisit oleh manusia. Dengan kata lain, komputer dapat mengamati data, mengekstraksi pola, dan membuat prediksi atau keputusan berdasarkan pola-pola yang ditemukan. Kajian ML berfokus pada pengembangan algoritma komputer dalam pengolahan data untuk mengatasi data yang besar dan kompleks serta menghasilkan suatu keluaran yang bersifat *intelligent action*.

Machine Learning menggunakan data yang berisi contoh dan fitur dari konsep yang akan dipelajari dan merangkum data ini dalam bentuk model yang kemudian digunakan untuk tujuan prediktif atau deskriptif. Model ini mampu mempelajari kumpulan data untuk dapat melakukan tugas tertentu yang sulit dibandingkan dengan program komputer biasa. Sebagai contoh, model ML dapat dengan mudah membedakan foto anjing dan kucing dalam sebuah gambar apabila dilatih dengan kumpulan data yang cukup banyak dibandingkan dengan program komputer biasa. Program berbasis ML melakukan prediksi berdasarkan variabel-variabel prediktor untuk mengekstrak pola dari kumpulan data sehingga sangat bergantung pada kualitas kumpulan data yang digunakan (Jana dan Biswas, 2021; Musolf dkk., 2022)

Proses pembelajaran ML dilakukan dengan menggunakan data yang dikenal sebagai data latih. Komputer menggunakan data ini untuk melakukan pelatihan agar memperoleh model. Proses pembelajaran ini memanfaatkan algoritma ML yang menerapkan teknik statistika. Fungsi model yang diperoleh tersebut digunakan untuk mengatasi berbagai permasalahan melalui proses *input-output* (Janiesch dkk., 2021). Proses pembelajaran dalam ML melibatkan tiga komponen utama:

1. Data

   Data adalah bahan bakar utama dalam ML. Komputer menggunakan data sebagai “pengalaman” untuk memahami dan memodelkan fenomena tertentu. Data dapat berupa berbagai jenis informasi, seperti gambar, teks, angka, atau bahkan suara.

2. Metrik dan Kesalahan.

   Metrik adalah ukuran kinerja yang digunakan untuk mengukur seberapa baik komputer telah "belajar" dalam melakukan prediksi atau membuat keputusan sesuai dengan kenyataan, sehingga algoritma ML dapat mengurangi tingkat kesalahan.

3. Umpan Balik (*Feedback*)

   Umpan balik adalah elemen kunci dalam pembelajaran. Pada saat komputer membuat kesalahan, umpan balik digunakan untuk memperbaiki model atau algoritma ML, sehingga kesalahan digunakan sebagai "pembelajaran" agar komputer dapat melakukannya dengan lebih baik di masa depan.

Menurut Lantz (2015), proses ML dibagi menjadi lima tahap sebagai berikut:

1. Mengumpulkan data.

   Pada tahap ini, data dapat diperoleh dari berbagai sumber selama data tersebut relevan untuk analisis yang dilakukan.

2. Mempersiapkan dan mengeksplorasi data.

   Kualitas data untuk proses ML merupakan hal yang sangat penting agar menghasilkan keluaran yang akurat. Langkah dalam proses ini membutuhkan campur tangan manusia. Sekitar 80% proses dikhususkan pada tahap ini, yang artinya sebagian besar waktu dihabiskan untuk mempelajari lebih lanjut tentang data. Selain itu, kumpulan data akan dibagi sesuai dengan rasio tertentu menjadi data pelatihan, data validasi, dan data pengujian.

3. Melatih model.

   Setelah data siap untuk dianalisis, algoritma dari ML tertentu akan merepresentasikan data dalam bentuk model.

4. Mengevaluasi performa data.

   Evaluasi model ML perlu dilakukan dengan data pengujian karena kemungkinan model yang dibangun dapat menyebabkan bias.

5. Meningkatkan performa data.

   Kinerja performa data harus ditingkatkan untuk mengembangkan ukuran kinerja, seperti menambah variabel dalam data, jumlah data dan sebagainya.

Beberapa jenis pekerjaan yang dapat diselesaikan dengan menggunakan model ML adalah sebagai berikut (Goodfellow dkk., 2016):

1. Klasifikasi.
   Pada jenis ini, model ML menentukan kelas atau kategori dari sebuah objek data berdasarkan masukkan pada model;
2. Regresi.
   Jenis ini mirip dengan klasifikasi, tetapi luaran yang dihasilkan bukan merupakan kelas atau kategori, tetapi angka yang kontinu;
3. Pengelompokan.
   Pada jenis ini, model ML menentukan kelompok dari sebuah objek data berdasarkan dengan kemiripan karakteristik antara objek data yang lain.

Data yang digunakan untuk membuat program berbasis ML biasanya dibagi menjadi data pelatihan, data validasi, dan data pengujian. Selama proses pelatihan dengan data pelatihan, *hyper-parameter* dari model disesuaikan agar model dapat melakukan prediksi dengan baik. Sedangkan, data validasi digunakan untuk memastikan bahwa performa model yang sedang dilatih tidak bias terhadap data pelatihan dengan membandingkan hasil prediksi model pada data pelatihan dengan data validasi. Setelah proses pelatihan, model yang telah dilatih diuji kembali dengan melakukan prediksi pada data pengujian yang tidak pernah digunakan sebelumnya.

Secara umum, berdasarkan ketersediaan informasi dari data, ML dapat dibedakan menjadi empat tipe algoritma (Sarker, 2021a), antara lain:

1. *Supervised Learning*
   Supervised learning merupakan salah satu pembelajaran mesin yang mampu menerima informasi atas dasar kumpulan data yang telah diberi label untuk diprediksi. Pada tipe ini, setiap observasi pada kumpulan data mempunyai variabel-variabel prediktor sebagai masukkan dan label atau target sebagai keluaran. Tujuan pembelajaran ini adalah mampu menghasilkan keluaran yang sesuai dengan pembelajaran masa lalu (target). Lebih lanjut, *supervised learning* dikelompokan ke dalam permasalahan klasifikasi atau regresi.

Variabel keluaran pada tugas klasifikasi berupa data kategorik, sementara variabel keluaran pada tugas regresi berupa data numerik. Kumpulan data yang diberi label digunakan untuk meringkas karakteristik distribusi masing-masing observasi dalam setiap jenis aplikasi, sehingga dapat dibentuk model prediksi atau klasifikasi data. Metode yang populer pada *supervised learning* antara lain adalah *Linear Regression* untuk regresi, *Logistic Regression* untuk klasifikasi, *Random Forest*, SVM, *k-Nearest Neighbour*, *Naïve Bayes*, *Neural Networks*, dan lain-lain.

2. *Unsupervised Learning*

   Unsupervised learning merupakan pembelajaran mesin yang digunakan pada data yang sebelumnya tidak memiliki informasi label atau target. Tujuan dari teknik pembelajaran ini adalah mengelompokkan data yang hampir sama dalam satu area tertentu dan tidak memerlukan keluaran target. *Unsupervised learning* sering disebut kluster atau pengelompokan dikarenakan tidak ada pemberian keterangan atau label dalam kumpulan data dan hasil yang diperoleh tidak mengidentifikasi suatu prediksi dari atribut label. Lebih lanjut, *unsupervised learning* dikelompokkan ke dalam permasalahan *clustering* dan *association*. *Clustering* adalah aturan yang digunakan untuk menemukan kelompok suatu objek berdasarkan observasi tersebut, sedangkan *association* adalah aturan yang menggambarkan hubungan antara observasi satu dengan observasi lainnya. Algoritma yang populer pada *unsupervised learning* adalah *k-means clustering*.

3. *Semi-Supervised Learning*

   Semi supervised learning merupakan gabungan teknik *supervised learning* dan *unsupervised learning*, karena berorientasi pada data yang memiliki label dan data yang tidak memiliki label. Data yang memiliki label sangat jarang terjadi dalam beberapa konteks di dunia nyata, dan data yang tidak memiliki label sangat banyak. Tujuan akhir dari *semi supervised learning* adalah memberikan hasil prediksi yang lebih baik daripada hasil prediksi menggunakan data yang telah memiliki label sehingga meningkatkan efisiensi hasil prediksi.

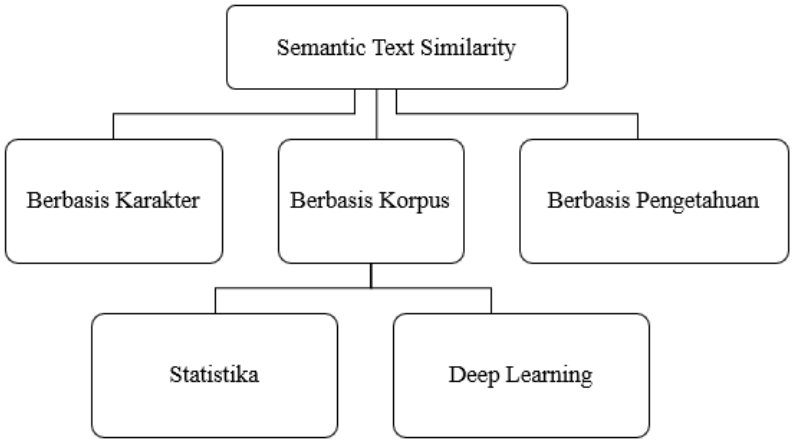
4. *Reinforcement Learning*

   Reinforcement learning adalah pendekatan berbasis lingkungan yang digunakan untuk mengevaluasi prediksi optimal dalam konteks atau lingkungan tertentu secara otomatis dalam meningkatkan efisiensinya. Pada tipe pembelajaran ini, algoritma model ML tidak hanya belajar dari kumpulan data, tetapi juga berinteraksi dengan lingkungan dengan tujuan tertentu. Pembelajaran ini didasarkan pada hasil atau resiko, yaitu algoritma ML memberikan *reward* (positif) saat model berprestasi baik dan sebaliknya mengurangi poin (negatif) saat model berprestasi buruk. Tujuan utamanya adalah menggunakan wawasan yang diperoleh dari aktivitas lingkungan untuk mengambil tindakan agar meningkatkan hasil dan meminimalkan resiko. Contoh penerapan model ini diaplikasikan pada mesin pencarian (*search*).

2.6 *Semantic Textual Similarity*

Semantic Textual Similarity (STS) merupakan kesamaan antara dua kalimat dalam teks berdasarkan makna kontekstualnya (Prasetya dkk., 2018). Pada bidang biomedis, misalnya, istilah “glukosa” dan “insulin” tampaknya tidak memiliki kesamaan, namun keduanya sangat terkait secara semantik karena merupakan istilah yang mengacu pada proses metabolisme. Tugas STS memainkan peran penting dalam beberapa tugas berbasis NLP, seperti temu kembali informasi (*information retrieval*), mesin penjawab otomatis (*automatic question answering*), mesin penerjemah (*machine translation*), *dialogue systems*, dan pencocokan dokumen (*document matching*) (Wang dan Dong, 2020).

Selama tiga dekade terakhir, berbagai pendekatan yang berbeda telah dikembangkan untuk mengukur kesamaan semantik antara satu teks dengan teks lainnya. Secara garis besar, pendekatan tersebut dibagi menjadi tiga kelompok seperti yang dideskripsikan pada Gambar 1, yaitu berdasarkan karakter (*string-based*), berdasarkan korpora (*corpus-based*), dan berdasarkan pengetahuan (*knowledge-based*) (Sunilkumar dan Shaji, 2019).



Gambar 1. Metode *Semantic Textual Similarity* (Sunilkumar dan Shaji, 2019).

Metode *string-based* atau berbasis karakter mengukur kesamaan dengan mempertimbangkan urutan karakter teks. Sementara, metode *corpus-based* atau berbasis korpora menghitung kemiripan dengan memanfaatkan korpora besar untuk menentukan hubungan antar kata yang menyusun teks. Metode ini terdiri dari dua pendekatan yang berbeda secara statistik, yaitu pendekatan yang melibatkan penggunaan teknik analisis statistik tradisional seperti LSA dan pendekatan yang melibatkan penggunaan model DL untuk menangkap konteks spesifik dalam teks. Di sisi lain, metode *knowledge-based* atau berbasis pengetahuan menggunakan informasi yang berasal dari domain pengetahuan untuk menghitung kesamaan semantik (Prakoso dkk., 2021).

2.7 *Cosine Similarity*

Cosine Similarity (CS) adalah ukuran jarak yang digunakan untuk mengukur kemiripan antara dua dokumen berdasarkan nilai kosinus antara dua vektor dokumen (Han dkk., 2012). *Cosine similarity* merupakan metrik yang diimplementasikan secara luas dalam pencarian informasi dan sering diterapkan untuk membandingkan kemiripan dua teks, seperti kalimat, paragraf atau seluruh dokumen (Rahutomo dkk., 2012).

Secara matematis, metode CS mengukur kosinus sudut antara dua vektor yang diproyeksikan dalam ruang multidimensi. Smentara secara khusus, metode CS mengukur kesamaan arah atau orientasi vektor dengan mengabaikan perbedaan besaran atau skalanya. Nilai suatu sudut kosinus yang menentukan kesamaan dua vektor mengandung nilai terkecil 0 dan nilai terbesar 1. Nilai 0 menandakan bahwa dokumen yang dibandingkan tidak ada kemiripan sama sekali dan apabila mendekati nilai 1 maka dokumen tersebut mempunyai kemiripan yang cukup besar (Lahitani dkk., 2016).

Cosine similarity memiliki dua keuntungan. Pertama, CS bermanfaat karena meskipun dua objek data serupa berjauhan berdasarkan jarak *Euclidean* karena ukurannya, tetapi sudut antara keduanya kemungkinan lebih kecil. Semakin kecil sudutnya, semakin tinggi kemiripannya. Kedua, saat di plot pada ruang multidimensi, CS menangkap sudut objek data dan bukan besarnya. Sudut yang lebih kecil antara vektor menghasilkan nilai kosinus yang lebih besar, menunjukkan tingkat kesamaan yang lebih besar. Sebagai contoh:

1. Ketika dua vektor memiliki sudut yang sama, sudut antara keduanya adalah 0, dan tingkat kesamaan kosinusnya adalah 1.
2. Vektor tegak lurus memiliki sudut 90 derajat antara keduanya dan kesamaan kosinusnya adalah 0.
3. Vektor yang berlawanan memiliki sudut 180 derajat di antara keduanya dan kesamaan kosinusnya adalah -1.

Nilai CS antar vektor dapat dihitung dengan Persamaan (1) sebagai berikut:

$$Cosine\ similarity\ (p, q) = \frac{(p.q)}{\|p\|.\|q\|} \quad (1)$$

Keterangan:

$\|p\|$ = panjang vektor p

$\|q\|$ = panjang vektor q

Dokumen dikatakan identik apabila sudutnya adalah nol derajat (0^o) dan kesamaannya adalah satu (1), dan dokumen itu dikatakan tidak identik apabila mempunyai sudut 90^o dan kesamaannya adalah nol (0).

2.8 *Word Mover's Distance*

Word Mover's Distance (WMD) adalah metrik jarak antara satu dokumen dengan dokumen lainnya (Pribadi dkk., 2021). *Word Mover's Distance* menghitung ketidaksamaan antara dua dokumen tekstual sebagai jarak minimum yang diperlukan untuk kata-kata yang ada dalam satu dokumen untuk dipindahkan agar sesuai dengan kata-kata yang ada di dokumen lainnya melalui Persamaan (2) sebagai berikut:

$$WMD(i,j) = \|x_i - x_j\|_2 \quad (2)$$

di mana:

WMD: *Word Mover's Distance*

x_i : Bobot dokumen

2.9 Klasifikasi

Klasifikasi adalah proses pengelompokan objek berdasarkan ciri-ciri yang dimilikinya, dan dalam konteks ini, fokus pada pengelompokan data atau konsep ke dalam kelas yang berbeda (Qamar dan Raza, 2020). Tujuan utama klasifikasi adalah menemukan model atau fungsi yang dapat menggambarkan dan membedakan kelas-kelas tersebut sehingga dapat digunakan untuk memprediksi kelas dari objek pengamatan yang belum diketahui.

Metode klasifikasi memiliki peran penting dalam analisis data dan pemahaman pola di dalamnya. Pada praktiknya, terdapat dua pendekatan umum dalam klasifikasi:

1. Pendekatan pertama adalah klasifikasi yang dilakukan secara manual oleh manusia tanpa melibatkan bantuan algoritma cerdas komputer. Pendekatan ini

sering digunakan pada saat kriteria klasifikasi sangat subjektif atau data tidak tersedia dalam jumlah besar.

2. Pendekatan kedua adalah klasifikasi yang menggunakan bantuan teknologi, khususnya algoritma ML. Beberapa algoritma ML, seperti *Naive Bayes*, SVM, *Decision Tree*, *Fuzzy Logic* dan *Artificial Neural Network* (ANN) digunakan untuk mengotomatisasi proses klasifikasi. Teknologi ini sangat berguna dalam menghadapi era big data, terutama dalam menghadapi volume data yang sangat besar yang membutuhkan analisis kuat dan efisien untuk mengungkap informasi berharga dan pengetahuan terorganisir. Penerapan metode ML dalam berbagai bidang termasuk klasifikasi telah menjadi semakin populer seiring dengan kemajuan teknologi kecerdasan buatan (Morimoto dkk., 2021).

Pada konteks statistika, metode klasifikasi yang umum digunakan adalah Analisis Diskriminan dan Regresi Logistik. Namun, pada era big data dan perkembangan teknologi kecerdasan buatan, metode ML seperti *Classification and Regression Tree* (CART), *Random Forest*, dan banyak lainnya menjadi solusi yang lebih efektif dalam melakukan klasifikasi data. Klasifikasi yang banyak digunakan secara umum terdapat 2 jenis, yaitu *binary classification* dan *multiclass classification*. *Binary classification* merupakan klasifikasi yang memisahkan dua buah kelas, sedangkan *multiclass classification* merupakan klasifikasi lebih dari dua kelas (Jha dkk., 2019). *Binary classification* memasukkan data ke dalam dua kelas yang berlawanan, seperti "Ya" dan "Tidak", "Positif" dan "Negatif". Sementara itu, *multiclass classification* digunakan apabila terdapat lebih dari dua kelas, misalnya klasifikasi jenis unggas ke dalam beberapa spesies yang berbeda.

Klasifikasi teks atau kategorisasi teks adalah metode pengelompokkan teks dalam *text mining*. Klasifikasi dalam *text mining* adalah proses menetapkan label ke unit tekstual seperti kalimat, *query*, paragraf, dan dokumen berdasarkan dokumen yang telah dilabeli sebelumnya (Korde, 2012; Zhao dkk., 2022). Teks adalah sumber informasi yang sangat kaya, tetapi proses ekstraksi wawasan dari teks merupakan kajian yang menarik dan membutuhkan waktu, karena sifat teks yang tidak terstruktur (Minaee dkk., 2021). Beberapa contoh penerapan klasifikasi teks, yaitu

menjawab pertanyaan, deteksi *spam*, analisis sentimen, kategorisasi berita, dan sebagainya. Data teks dapat berasal dari berbagai sumber, termasuk data *web*, *email*, obrolan, media sosial, tiket, klaim asuransi, ulasan pengguna, dan sebagainya. Masalah yang sering muncul dalam klasifikasi adalah data yang tidak seimbang, dimana kelas minoritas berjumlah lebih sedikit daripada kelas mayoritas.

2.10 *Imbalanced* Data

Imbalanced data merupakan kumpulan data yang memiliki perbedaan signifikan dalam jumlah data antar kelompok atau kelas atau label atau target. Kelompok dengan jumlah data yang lebih banyak disebut sebagai mayoritas, sementara kelompok dengan jumlah data yang lebih sedikit disebut sebagai minoritas. Distribusi jumlah data pada setiap kelas tidak seimbang akan berdampak buruk pada model yang dibangun, karena algoritma cenderung lebih banyak memperhatikan kelas mayoritas dibandingkan kelas minoritas, sehingga *imbalanced* kumpulan data dapat mempengaruhi kinerja model (Soltanzadeh dkk., 2023).

Imbalanced data atau ketidakseimbangan data merupakan salah satu permasalahan klasifikasi yang sangat krusial pada ML, karena rasio data tidak seimbang dapat mengakibatkan model kesulitan dalam melakukan klasifikasi data (Hayashi dan Fujita, 2022). Hal ini terjadi pada saat kelas minoritas jauh lebih kecil dari kelas mayoritas (Ren dkk., 2017). Kondisi tersebut cenderung menyebabkan keluaran dari model klasifikasi yang dibangun bias terhadap kelas mayoritas, sehingga kelas minoritas diabaikan. Kondisi ini sangat tidak disukai, karena terkadang kelas minoritas merupakan kelas yang memiliki informasi penting (Shaikh dkk., 2021b).

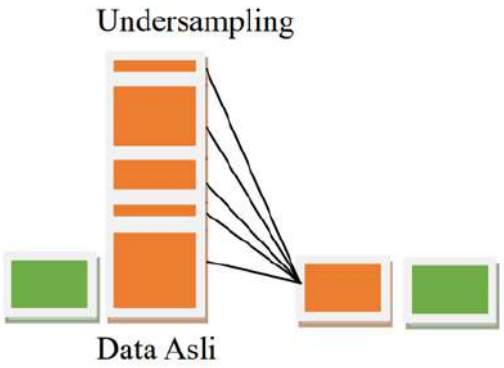
Model yang dibangun berdasarkan data tidak seimbang akan menghasilkan akurasi prediksi minoritas yang rendah, karena informasi yang kaya dari mayoritas mendominasi minoritas sehingga menyebabkan batas-batas keputusan yang bias dalam sistem klasifikasi (Jian dkk., 2016). Ketidakseimbangan data telah menjadi subjek penelitian lebih dari satu dekade. Japkowicz dan Stephen (2002) dalam eksperimennya menunjukkan bahwa tingkat kerumitan masalah ketidakseimbangan

data tergantung pada kompleksitas tugas, rasio ketidakseimbangan data, dan jumlah data yang digunakan pada proses pelatihan. Semakin tinggi ketidakseimbangan data, semakin buruk masalah bias terhadap kelas mayoritas (Liu dkk., 2019). Selain itu, ketidakseimbangan data dan *noise* juga dapat berpengaruh pada kualitas kinerja model klasifikasi.

2.11 *Resampling*

Resampling adalah salah satu teknik *preprocessing* dimana distribusi data diseimbangkan untuk mengurangi efek distribusi kelas tidak seimbang dalam proses pembelajaran (Jian dkk., 2016). *Resampling* secara luas digunakan untuk menyelesaikan masalah data yang tidak seimbang. Teknik ini dilakukan dengan cara menyamakan jumlah data asli berdasarkan algoritma *sampling* yang menyesuaikan jumlah data dalam kelas yang berbeda, serta melatih data "seimbang" baru dengan mengadopsi algoritma klasifikasi.

Pendekatan *resampling* dibagi menjadi tiga, yaitu metode *undersampling*, *oversampling*, dan *hybrid* yang menggabungkan metode *undersampling* dan *oversampling* (Jian dkk., 2016). Metode *undersampling* merupakan salah satu cara mengatasi ketidakseimbangan data dengan mengeliminasi data dari kelas mayoritas. Beberapa variasi populer dari *undersampling* adalah *Random Undersampling* (RUS), dan *Tomek's Link Undersampling*. Pada metode *undersampling*, beberapa data pada kelas mayoritas dihilangkan sehingga terdapat kemungkinan hilangnya data berharga pada kelas mayoritas. Metode ini sangat baik apabila diterapkan pada kumpulan data yang memiliki banyak observasi minor. Gambar 2 merupakan ilustrasi grafik dari *undersampling*:



Gambar 2. Grafik *Undersampling* (Mohammed dkk., 2020).

Metode *oversampling* merupakan salah satu cara untuk mengatasi masalah ketidakseimbangan data dengan cara mereplikasi sampel kelas minoritas melalui beberapa variasi teknik *oversampling*, seperti *Random Oversampling* (ROS), *Synthetic Minority Oversampling* (SMOTE), *Borderline* SMOTE, dan *Support Vector Machine* SMOTE. Metode ini bertujuan menghasilkan sebuah kumpulan data pelatihan baru dari replikasi kelas minoritas untuk menyeimbangkan jumlah data (He dan Ma, 2013; He dkk., 2018; Rajesh dan Dhuli, 2018; Wongvorachan dkk., 2023). *Oversampling* divisualisasikan pada Gambar 3 sebagai berikut:

Gambar 3. Grafik *Oversampling* (Mohammed dkk., 2020).

Metode terakhir adalah metode *hybrid* yang menggabungkan metode *undersampling* dan *oversampling*. Metode tersebut merupakan teknik *resampling* yang dilakukan dengan cara mengurangi jumlah data pada bagian kelas mayoritas sedemikian rupa sehingga proporsi kelas mayoritas tidak sebesar proporsi aslinya,

kemudian mereplikasi kelas minoritas sedemikian rupa sehingga proporsi jumlah data lebih besar dibandingkan proporsi aslinya. Kombinasi kedua teknik *resampling* tersebut baik untuk digunakan jika ukuran data latih terlalu besar (Haixiang dkk., 2017).

Berdasarkan temuan Pristyanto dkk. (2018) dan Hassan dkk (2020), metode *resampling* yang berhasil untuk mengatasi masalah ketidakseimbangan data pada kasus *multiclass* adalah metode RUS dan metode ROS. Namun, untuk memperoleh hasil yang lebih optimal, penelitian ini akan menggabungkan kemampuan RUS dan ROS bersama-sama, sehingga jumlah kelas mayoritas dikurangi dan jumlah kelas minoritas ditingkatkan pada saat yang sama. Tahapan pertama yang akan dilakukan adalah menerapkan metode RUS dengan menghapus atau mengurangi data dari kelas mayoritas secara acak berdasarkan prosedur seleksi pada Persamaan (3) berikut (Wongvorachan dkk., 2023):

$$x_{rus} = (x_{ma} - x_i) * \delta \tag{3}$$

dengan x_{rus} adalah sampel yang telah di *undersampling*, x_{ma} adalah sampel kelas mayoritas, x_i adalah salah satu tetangga terdekat dari sampel mayoritas, dan δ adalah angka acak yang dihasilkan antara 0 dan 1.

Apabila data dari kelas mayoritas telah dikurangi, langkah selanjutnya adalah mereplikasi kelas minoritas secara acak menggunakan metode ROS hingga jumlah data pada kedua kelas seimbang. Prosedur replikasi data yang digunakan oleh metode ROS diuraikan pada Persamaan (4) sebagai berikut:

$$x_{new} = x_{mi} + (x_i - x_{mi}) * \delta \tag{4}$$

dengan x_{new} adalah sampel yang di *generate*, x_{mi} adalah sampel kelas minoritas, x_i adalah salah satu tetangga terdekat dari sampel minoritas, dan δ adalah angka acak yang dihasilkan antara 0 dan 1.

2.12 *Word Embedding*

Bengio dkk. (2003) memperkenalkan istilah *word embedding* pertama kali dan mendefinisikannya sebagai sebuah fungsi parameter yang memetakan setiap kata ke dalam vektor berdimensi tinggi sebagai proses konversi kata yang berupa karakter *alphanumeric* ke dalam bentuk *vektor*. Secara terminologi *word embedding* disebut juga dengan representasi vektor kata, yang merupakan teknik pembelajaran dalam NLP untuk mengubah kata atau frasa ke dalam bentuk vektor (Jatnika dkk., 2019). Representasi kata ini memiliki dampak yang sangat penting terhadap performa dan akurasi model pembelajaran yang dibangun (Asudani dkk., 2022).

Salah satu algoritma klasik, yaitu *one-hot encoding* merupakan metode standar data kategorikal untuk melakukan konversi kata atau teks ke dalam format vektor. Metode ini bekerja dengan cara membuat *array* satu dimensi yang memiliki panjang sebanyak jumlah fitur yang digunakan. *Array* yang dibangun berupa bilangan biner dengan rentang nilai antara 0 sampai 1. Nilai 0 adalah representasi nilai kategori yang tidak memiliki hubungan sekuensial, sedangkan nilai 1 adalah sebaliknya (Vijayalakshmi dan Venkatachalapathy, 2019). *One hot encoding* dapat menjelaskan data bertipe kategorikal secara lebih ekspresif. Namun, algoritma ini terlalu sederhana dan tidak memperhitungkan konteksnya. Berbagai algoritma representasi kata telah ditemukan, seperti teknik *Bag of Words* (BOW), *bi-gram*, *n-gram*, dan *Term Frequency - Inverse Document Frequency* (TF-IDF).

Menurut Mikolov dkk. (2013b) dan Mnih dan Kavukcuoglu (2013), representasi vektor kata ke dalam ruang *low-dimensional word embedding* jauh lebih cocok untuk model terbaru DL berbasis *neural network*. Pendekatan ini mewakili setiap kata sebagai vektor *dense* bilangan riil, di mana kata-kata yang secara semantik terkait satu sama lain memetakan ke vektor serupa. Sehingga, representasi vektor kata tersebut dapat menangkap makna semantik yang berguna dan hubungan linguistik antara kata-kata menggunakan *deep neural network* (Mikolov dkk., 2013c; Levy dan Goldberg, 2014). Di antara pendekatan *neural embedding*, model *skip-gram* telah mencapai hasil terbarukan dalam banyak tugas NLP, termasuk

penyelesaian kalimat, analogi, dan analisis sentimen (Mikolov dkk., 2013a; Mikolov dkk., 2013b; Fernández dkk., 2014).

Terdapat beberapa peningkatan jumlah penelitian yang menerapkan *word embedding* dalam tugas NLP umum, seperti bidang *Information Extraction* (IE) (Nguyen dan Grishman, 2014; Zeng dkk., 2014; Wang dkk., 2017), bidang *Information Retrieval* (IR) (Ganguly dkk., 2015), bidang analisis sentimen (Maas dkk., 2011; Tang dkk., 2014), bidang otomasi penjawab pertanyaan (Dong dkk., 2015; Ren dkk., 2015), dan bidang ringkasan teks (Rush dkk., 2015; Yogatama dkk., 2015). Kajian tentang *word embedding* pada domain biomedis juga telah digunakan secara luar biasa dalam berbagai aplikasi seperti *medical synonym extraction* (Jagannatha dkk., 2015), bidang *Extraction Relation* (ER) dalam aplikasi *chemical-disease relation* (Jiang dkk., 2015), interaksi obat-obat (Liu dkk., 2016; Wang dkk., 2017), interaksi protein-protein (Jiang dkk., 2016), dan IR biomedis (Wang dkk., 2016; Malik dkk., 2022), serta disambiguasi singkatan *medical* (Wu dkk., 2015).

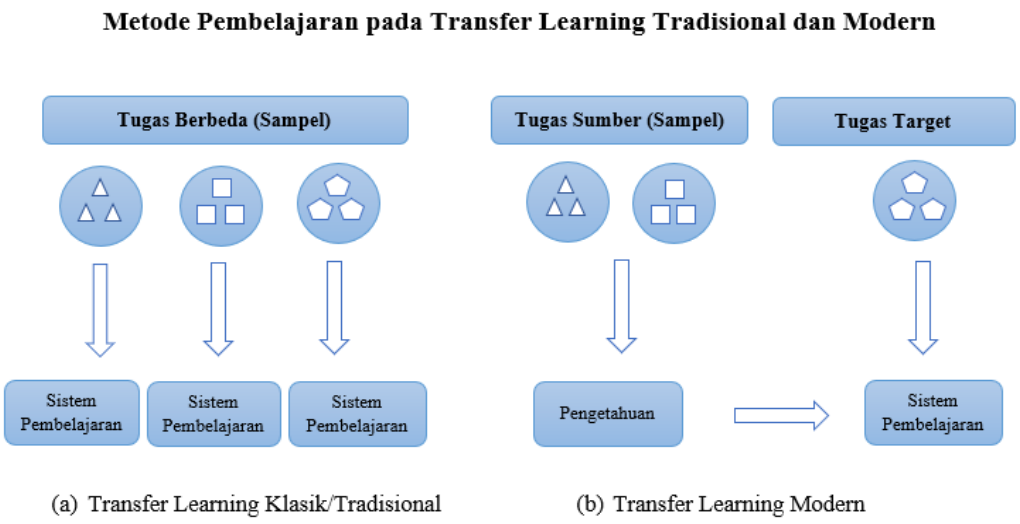
Ada dua sumber teks utama yang digunakan untuk melatih *word embedding* pada aplikasi NLP biomedis, yaitu korpora tugas internal, seperti data latih yang dilakukan oleh Srinivasan dkk. (2021) dan sumber data eksternal, seperti Wikipedia yang dilakukan oleh Gurulingappa dkk. (2016). *Word embedding* yang diperoleh dari data latih disebut *custom-trained embedding*, sedangkan *word embedding* yang diperoleh dari sumber data eksternal disebut *pre-trained word embedding*. Pemanfaatan sumber data eksternal didasarkan pada asumsi implisit bahwa sumber eksternal berisi pengetahuan yang dapat digunakan untuk meningkatkan tugas domain (Shen dkk., 2015; Shen dkk., 2016; Shen dan Lee, 2016).

2.13 *Transfer Learning*

Sejak awal era ML, *Transfer Learning* (TL) telah menjadi penelitian yang menarik bagi para ilmuwan. Pada era tersebut, TL dikenal sebagai adaptasi domain dan berfokus pada pemrosesan dan integrasi kumpulan data homogen karena model ML

tradisional tidak bergantung pada ukuran himpunan data yang digunakan untuk proses pelatihan.

Sementara pada era DL, meskipun telah berhasil menangani sejumlah aplikasi yang menantang dan menarik dalam beberapa tahun terakhir; seperti *image processing*, dan NLP, model DL juga memiliki beberapa kelemahan, misalnya; membutuhkan data yang memiliki label dan proses pelatihan yang mahal (waktu dan biaya). Kendala tersebut ternyata dapat diatasi dengan metode TL terbaru, di mana pengetahuan atau masukkan yang telah dilatih sebelumnya akan di *transfer* ke domain atau tugas baru yang berbeda. Metode ini memungkinkan model yang dibangun untuk memulai dari pola yang telah dipelajari sebelumnya untuk memecahkan masalah yang berbeda, sehingga dapat mengurangi waktu dan biaya proses pelatihan, serta kebutuhan akan kumpulan data pelatihan yang memiliki label yang mungkin sulit diperoleh di beberapa bidang seperti biomedis. Selain itu, metode terbaru TL juga mampu menangani masalah *overfitting*, yang berdampak pada peningkatan kinerja model. Oleh karena itu, motivasi untuk menggunakan TL pada era DL lebih tinggi dari era sebelumnya (Rawat dan Wang, 2017; Iman dkk., 2023; Kotei dan Thirunavukarasu, 2023). Perbedaan metode TL pada era ML dan DL diilustrasikan pada Gambar 4 sebagai berikut:



Gambar 4. Perbedaan Metode *Transfer Learning* (Hosna dkk., 2022).

Menurut Zhang dan El-Gohary (2021), TL dapat dikategorikan menjadi tiga jenis berdasarkan bagaimana pengetahuan di *transfer* dari domain sumber ke domain target, yaitu:

1. *Instance-based*

   Data domain dari sumber akan di *resampling* agar serupa dengan data pada domain target (misalnya, menerapkan metode *boosting* pada klasifikasi teks lintas domain).
2. *Feature-based*

   Feature-based menggunakan fitur atau representasi kata yang dapat di *transfer* melalui model DL (misalnya, *global vectors* untuk model *word embedding*)
3. *Model-based*

   Model-based menerapkan kembali sebagian *deep neural network* (lapisan yang dilatih pada domain sumber) ke domain target dengan memperbarui model menggunakan data domain target. Contoh teknik adaptasi *model-based* mencakup *fine tuning* (penyempurnaan) pada model klasifikasi gambar berbasis *pre-trained* CNN (misalnya: *VGG19, Inceptionv3, ResNet50*), dan *fine tuning* pada model berbasis *Transformer* (misalnya: GPT, dan BERT) untuk tugas analisis teks tertentu.

Berdasarkan pengelompokkan di atas, pengetahuan yang di *transfer* pada metode *feature-based* ialah berupa *word embedding.* Terdapat dua macam *word embedding* berdasarkan cara memperolehnya yaitu *custom-trained embedding* dan *pre-trained word embedding.* Pada *custom-trained embedding,* vektor yang dihasilkan hanya dapat digunakan untuk tugas spesifik di mana *word embedding* dihasilkan. Sedangkan, vektor representasi pada *pre-trained word embedding* dapat di *transfer* pada tugas-tugas lain karena dilatih pada kumpulan data besar, dan disimpan. Sehingga dapat disimpulkan bahwa aplikasi *pre-trained word embedding* pada model DL merupakan salah satu bentuk TL dengan konsep *feature-based*.

2.14 *Hyperparameter Tuning*

Proses membangun model DL yang baik membutuhkan waktu serta *effort* yang besar karena melibatkan pemilihan algoritma dan penyetelan parameter yang tepat untuk menciptakan arsitektur model yang ideal. Yang dan Shami (2020) membagi parameter dalam model DL menjadi dua jenis, yaitu:

1. Parameter model

   Parameter yang dapat diinisialisasi dan diperbarui melalui proses pembelajaran (misalnya, bobot *neuron* di *neural network*).
2. *Hyperparameter*

   Parameter yang nilai estimasinya harus ditetapkan terlebih dahulu sebelum proses pelatihan. *Hyperparameter* dapat berupa parameter yang digunakan untuk mengkonfigurasi model DL, seperti *learning rate*, atau untuk meminimalkan fungsi kerugian, seperti fungsi aktivasi dan *dropout*.

Proses perancangan arsitektur model ideal dengan kombinasi *hyperparameter* yang tepat disebut *hyperparameter tuning*. Menurut Li dkk. (2021), kombinasi *hyperparameter* yang tepat dapat meningkatkan kinerja model DL secara keseluruhan. Pada praktiknya, kombinasi tersebut dapat diperoleh dengan melakukan sejumlah percobaan manual (*trial and error*). Namun dengan meningkatnya kompleksitas model yang memiliki lebih banyak *hyperparameter*, upaya menghitung semua kombinasi *hyperparameter* secara manual menjadi tidak efektif lagi. Hal ini dikarenakan metode ini membutuhkan biaya yang mahal sementara hasil yang diperoleh tidak selalu mampu meningkatkan kinerja model. Oleh karena itu, berbagai metode optimasi seperti *grid search* dan *random search* dikembangkan sebagai alternatif dalam penentuan *hyperparameter* secara otomatis agar model DL dapat menghasilkan performa terbaik dengan metode yang lebih efisien.

Metode *grid search* adalah metode yang mengkombinasikan semua nilai *hyperparameter* yang mungkin digunakan untuk dilatih dan dievaluasi oleh model. Hasil yang diperoleh kemudian akan dibandingkan untuk memilih kombinasi *hyperparameter* yang memberikan kinerja terbaik. Meskipun implementasinya

mudah dan telah digunakan secara luas untuk *hyperparameter tuning*, metode tersebut bukan merupakan metode yang efisien untuk menentukan *hyperparameter* dengan dimensi yang lebih tinggi.

Bergstra dan Bengio (2012) mengusulkan *random search* untuk *hyperparameter tuning*. Metode *random search* disebut juga sebagai metode optimasi acak karena kombinasi nilai *hyperparameter* diperoleh secara acak dari sampel yang mewakili populasi tanpa perlu mencoba setiap kombinasi. Dengan jumlah percobaan yang sama, *random search* dapat menghasilkan kombinasi *hyperparameter* yang sama baiknya dengan *grid search*. Oleh karena itu, *random search* adalah metode yang lebih efisien dalam menentukan *hyperparameter* dibandingkan *grid search.*

2.15 *Siamese Neural Network*

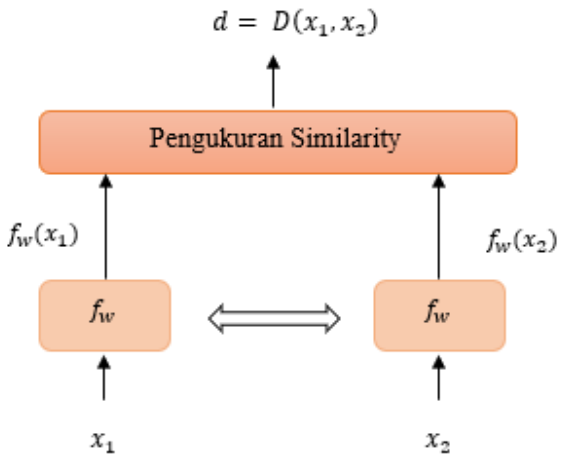
Siamese Neural Network atau biasa disebut sebagai *Twin Neural Network* merupakan ANN yang terdiri dari dua *sub-network* identik, dengan bobot dan parameter yang sama (Serrano dan Bellogín, 2023). Tidak seperti arsitektur ANN lainnya, *Siamese Neural Network* tidak mudah mengalami *overfitting* dan hanya memerlukan sedikit data pelatihan. Selain itu, implementasi *Siamese Neural Network* memungkinkan untuk mengkonfigurasi jaringan sesuai dengan tugas, termasuk penggunaan model DL seperti CNN dan LSTM (de Souza dkk., 2020).

Algoritma *Siamese Neural Network* pertama kali diusulkan oleh Bromley dkk. (1994) untuk mendeteksi tanda tangan palsu. Namun, Baldi dan Chauvin (1993) pernah memperkenalkan ANN serupa yang mampu mengenali sidik jari dengan nama berbeda. Pada penelitian oleh Bromley dkk. (1994), *Siamese Neural Network* berhasil memverifikasi keaslian tanda tangan dengan cara mengukur kesamaan antara dua tanda tangan.

Sejak kemunculannya, model *Siamese Neural Network* telah diaplikasikan pada berbagai bidang termasuk *text mining*. Akan tetapi, tidak ada studi atau penelitian yang mengintegrasikan *Siamese Neural Network* dengan *text mining* sampai lebih

dari satu dekade. Penelitian ini diusulkan kembali oleh Yih dkk. (2011) untuk mempelajari kesamaan pada kumpulan data Wikipedia.

Secara sederhana, *Siamese Neural Network* melatih fungsi f_w untuk memetakan sepasang data masukkan x_1 dan x_2 ke dalam ruang dimensi yang lebih kecil sehingga jarak d antara vektor fitur $f_w(x_1)$ dan $f_w(x_2)$ nilainya mendekati 1 untuk data masukkan yang serupa secara semantik, dan nilainya mendekati 0 untuk data masukkan yang berbeda (Chopra dkk., 2005). Skema jaringan *Siamese* direpresentasikan pada Gambar 5 sebagai berikut:



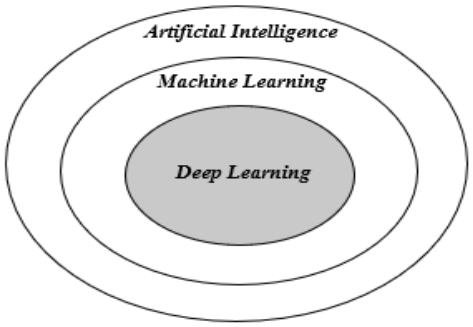
Gambar 5. Arsitektur Sederhana *Siamese Neural Network* (Ilina dkk., 2022).

Fungsi pemetaan f_w di masing-masing jaringan *sub-network* pada gambar di atas adalah representasi dari model DL yang paling umum digunakan untuk mengukur kesamaan, yaitu model CNN (Kim, 2014; Shao, 2017), model LSTM (Tien dkk., 2019), atau model *Bidirectional*-LSTM (He dan Lin, 2016). Sedangkan $d = D(x_1, x_2)$ adalah keluaran dari jaringan *Siamese* yang dihitung berdasarkan metrik jarak antara pemetaan data masukkan $f_w(x_1)$ dan $f_w(x_2)$. Metrik jarak yang digunakan dapat berupa jarak *Euclidean*, *Cosine*, atau *Manhattan* (Atanbori dan Rose, 2022).

2.16 *Deep Learning*

Pada akhir tahun 1980-an, *neural network* telah menjadi topik utama di bidang ML dan *Artificial Intelligence* (AI) karena penemuan berbagai metode pembelajaran dan struktur jaringan yang berhasil digunakan di banyak aplikasi (Karhunen dkk., 2015). Namun dari tahun ke tahun, minat untuk meneliti topik ini mengalami penurunan.

Pada tahun 2006, Hinton dkk. (2006) memperkenalkan model DL yang didasarkan pada konsep *neural network*. *Deep learning* kemudian menjadi topik penting yang mengarah pada kebangkitan penelitian terkait *neural network*. Saat ini, DL menjadi salah satu topik penelitian yang menarik di bidang ML, AI, serta *Data Science and Analytic* karena kemampuan untuk mempelajari data yang diberikan. Berdasarkan domain kerja, seperti yang ditampilkan pada Gambar 6, DL adalah bagian dari ML dan AI, sehingga DL dapat disebut sebagai kecerdasan buatan yang meniru otak manusia untuk pemrosesan data (Sarker, 2021b).



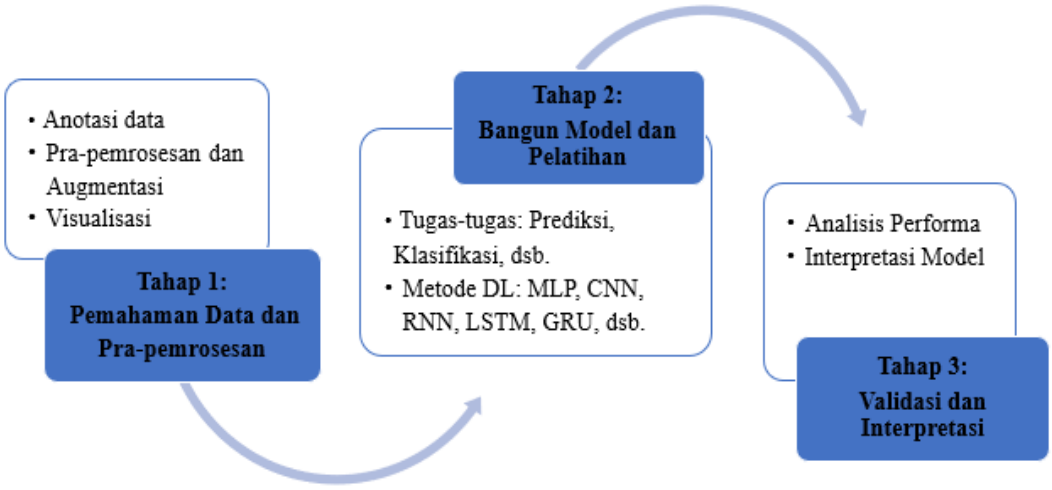
Gambar 6. Ilustrasi Posisi *Deep Learning* (Sarker, 2021b).

Pada aplikasi dunia nyata, data yang digunakan untuk model DL dapat ditemukan dalam berbagai bentuk, yang dapat direpresentasikan sebagai berikut:

1. Data sekuensial: contoh data sekuensial adalah teks, fragmen audio, klip audio, dan deret waktu. Saat membangun model dengan data sekuensial, urutan dari data masukkan harus diperhatikan secara eksplisit.
2. Gambar atau data dua dimensi (2D): gambar digital yang terdiri dari susunan angka, atau simbol pada baris dan kolom dalam matriks 2D.

3. Data tabular: kumpulan data dalam format kolom seperti pada tabel. Secara keseluruhan merupakan susunan data yang logis dan sistematis dalam bentuk baris dan kolom yang didasarkan pada sifat atau fitur data.

Metode pemrosesan data pada model DL terdiri dari tiga tahapan, yaitu: memahami dan mengolah data, membuat dan melatih model DL, serta validasi dan interpretasi model. Metode tersebut hampir sama dengan model ML tradisional. Meskipun demikian, untuk sebagian besar aplikasi yang membutuhkan pemrosesan data teks, gambar, video, ucapan, dan audio, model DL terbukti lebih efektif daripada model ML tradisional (LeCun dkk., 2015). *K-nearest neighbor*, SVM, *Decision Tree*, *Random Forest*, *Naive Bayes*, *Linear Regression*, *Association Rules*, dan *K-means Clustering*, adalah beberapa contoh model ML tradisional yang umum digunakan di berbagai domain aplikasi. Di sisi lain, model DL mencakup model-model yang menerapkan *multi layer neural network* seperti *Multi Layer Perceptron* (MLP), CNN, RNN, dan *Deep Belief Network.* Gambar 7 merupakan alur kerja pada model DL:



Gambar 7. Alur Kerja *Deep Learning* (Sarker, 2021b).

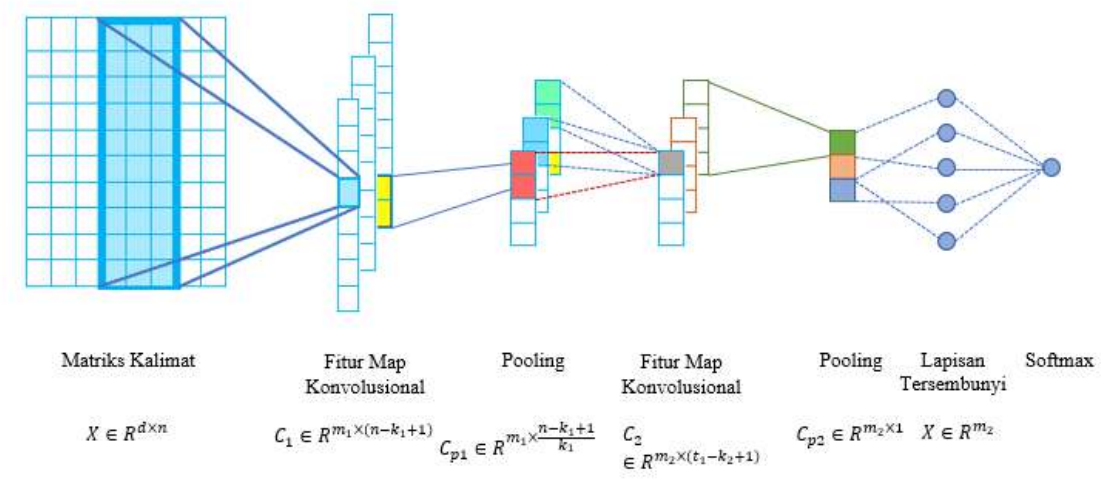
Salah satu ciri khas DL adalah selain memiliki lapisan *input* dan lapisan *output*, model ini juga terdiri dari beberapa lapisan tersembunyi (lapisan tersembunyi ≥ 2) dan umumnya berisi *neuron* tingkat lanjut (misalnya; konvolusi pada CNN) yang

berbeda dengan ANN sederhana. Karakteristik tersebut memungkinkan DL mencapai kinerja yang lebih baik daripada ML, bahkan saat memproses data dalam jumlah besar (Janiesch dkk., 2021).

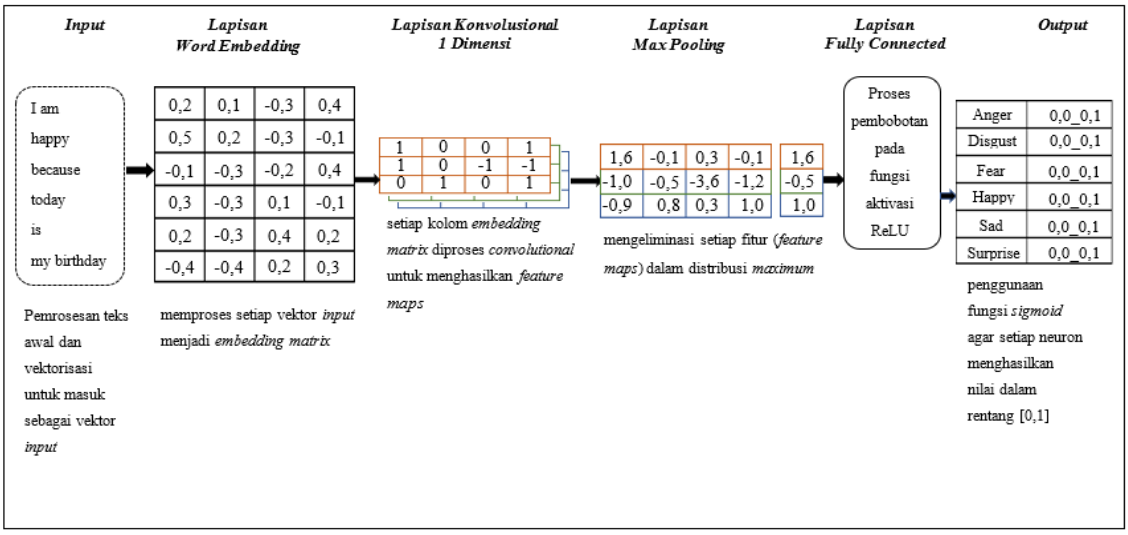
2.17 *Convolutional Neural Network*

Convolutional Neural Network adalah algoritma yang paling terkenal dan sering digunakan. Keunggulan utama CNN adalah mampu secara otomatis mengidentifikasi fitur-fitur yang relevan tanpa adanya pengawasan manusia. Implementasi CNN telah banyak diaplikasikan pada berbagai bidang, seperti *computer vision*, deteksi objek, segmentasi gambar, pemrosesan bahasa alami, pengenalan ucapan, dan pengenalan pola hingga pengembangan teknologi kendaraan otonom. Struktur CNN terinspirasi oleh susunan sel-sel otak manusia dan hewan, yang mirip dengan jaringan saraf konvensional. Model CNN memiliki tiga manfaat utama, yaitu representasi yang setara, interaksi yang jarang, dan pembagian parameter.

Secara umum, CNN merupakan model ANN yang memberi umpan balik dimana jaringan saraf mempertahankan struktur hierarki dengan mempelajari representasi fitur internal dan menggeneralisasi fitur-fitur. Penggunaan model CNN tidak terbatas pada gambar, tetapi dapat diterapkan pada teks, suara, dan video. Arsitektur CNN terdiri dari beberapa lapisan yang berurut. Lapisan pertama adalah lapisan konvolusi, yang bertugas mendeteksi berbagai fitur. Lapisan kedua adalah lapisan *sub-sampling* atau *pooling*, yang berfungsi untuk mereduksi resolusi fitur dan mengurangi jumlah parameter yang digunakan. Lapisan terakhir adalah lapisan *fully connected*, yang berfungsi untuk menggabungkan hasil-hasil dari lapisan sebelumnya dan menghasilkan keluaran akhir berupa klasifikasi. Arsitektur CNN divisualisasikan pada Gambar 8a dan Gambar 8b sebagai berikut:



Gambar 8a. Arsitektur *CNN* (Deriu dkk., 2017).



Gambar 8b. Arsitektur *CNN* (Aripin dkk., 2021).

Pada bidang NLP, metode DL dan *neural network* telah banyak dikembangkan dalam ML (Bengio, 2009; LeCun dkk., 2015). Metode-metode tersebut terbukti berhasil untuk tugas pemrosesan gambar dan ucapan. Metode *neural network* mulai mengambil alih model linier tradisional yang jarang diterapkan untuk NLP (Bengio dkk., 2003; Collobert dan Weston, 2008; Collobert dkk., 2011; Mikolov, dkk., 2013b; Socher dkk., 2013). Metode DL telah terbukti handal dalam sejumlah permasalahan klasifikasi. Salah satunya, CNN mampu secara efisien menangkap representasi bermakna dari kalimat seperti pada klasifikasi dan pemodelan bahasa

(Kalchbrenner dkk., 2014; Iyyer dkk., 2015; Goldberg, 2016), analisis sentimen (Kim, 2014), hingga ekstraksi informasi (Nguyen dan Grishman, 2014; Chen dkk., 2015; Nurdin dan Maulidevi, 2018).

Model CNN memanfaatkan representasi terdistribusi dari kata-kata dengan terlebih dahulu mengubah kata-kata dalam setiap kalimat menjadi vektor, kemudian membentuk matriks yang digunakan sebagai masukkan. Penggunaan *word embedding* pada arsitektur CNN untuk klasifikasi kalimat banyak digunakan oleh para peneliti di antaranya, Kalchbrenner dkk. (2014) mengusulkan arsitektur CNN dengan skema *pooling* yang kompleks dan memperoleh kinerja yang baik tanpa memerlukan fitur eksternal. Sementara, Kim (2014) mengusulkan CNN sederhana dengan menggunakan satu lapisan konvolusi dan *static vectors* (Word2Vec) memperoleh hasil yang kompetitif dibandingkan model DL yang menerapkan skema *pooling* yang kompleks.

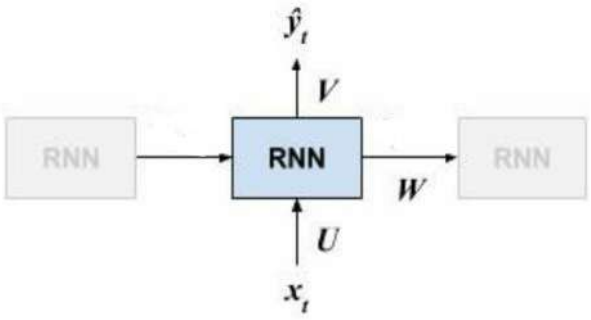
Model CNN memiliki tingkat akurasi yang lebih tinggi pada beberapa penelitian yang dilakukan pada pengenalan karakter. Sebagai contoh, Kim dan Xie (2015) melakukan penelitian untuk pengenalan karakter pada tulisan tangan Korea dari 2 *database* berbeda dan memperoleh nilai akurasi tertinggi, yaitu 99,71%. Notonogoro dkk. (2018) juga melakukan penelitian tentang pengenalan karakter plat kendaraan Indonesia dan akurasi yang diperoleh sebesar 95,45%. Dari kedua penelitian tersebut, CNN digunakan untuk melakukan identifikasi atau klasifikasi kata pada citra teks. Cara kerja CNN yang hampir mirip dengan MLP diyakini cenderung lebih efektif dan mudah dilatih, karena CNN melakukan konvolusi pada masukkan untuk mereduksi ukuran citra (matriks).

2.18 *Long Short-Term Memory*

Pada tahun 1997, metode LSTM diperkenalkan pertama kali oleh Hochreiter dan Schmidhuber kemudian dikembangkan oleh Gers dkk. (2000). Namun penelitian ini telah disempurnakan pada penggunaan skala besar praktis oleh Cho dkk. (2014). Sampai saat ini, pengembangan penelitian metode LSTM masih terus

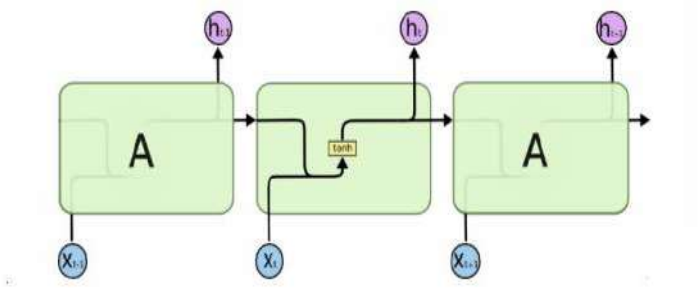
dikembangkan oleh para peneliti, sehingga model LSTM telah mencapai performa terbaik dalam tugas AI pada berbagai bidang, seperti pada *handwriting recognition* dan *natural speech recognition* (Graves dkk., 2009, Graves dkk., 2013), kemudian pada bidang *robotic control* (Zhao dkk., 2019), dan bidang *automated game control* (Bouzidi dan Hashemi, 2023). Pada tahun 2016, perusahaan teknologi besar termasuk Google, Apple, Facebook, Amazon, dan Microsoft mulai menggunakan LSTM sebagai komponen inti dalam *speech recognition* dan sistem penerjemah bahasa otomatis.

Long Short-Term Memory merupakan salah satu arsitektur alternatif *Recurrent Neural Network* (RNN) yang dibangun untuk dapat mengelola data dalam skala besar dan menangani kelemahan RNN dalam menyimpan memori jangka panjang (Fan dkk., 2020). *Long Short-Term Memory* memiliki perbedaan dengan RNN pada struktur modul pengulangan saja. Namun secara fundamental, LSTM tidak memiliki perbedaan arsitektur dari RNN yang ditampilkan pada Gambar 9 sebagai berikut:



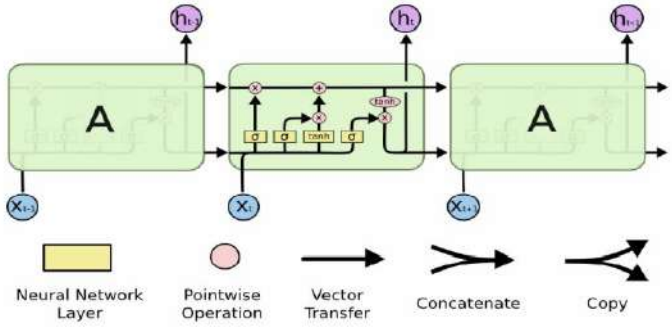
Gambar 9. Arsitektur RNN (Firmansyah dkk., 2020).

Arsitektur RNN memiliki bentuk rantai modul pengulangan ANN dengan struktur yang sangat sederhana, seperti lapisan $tanh$ tunggal. Arsitektur tersebut ditampilkan pada Gambar 10 sebagai berikut:



Gambar 10. Modul Pengulangan RNN yang berisi satu *layer*
(Alakkari dkk., 2023).

Recurrent Neural Network memiliki kekurangan, kekurangan itu dapat dilihat pada masukkan x_{t-1}, x_t yang memiliki rentang informasi yang sangat besar dengan x_t, x_{t+1} . Sehingga ketika h_{t+1} memerlukan informasi yang relevan dengan x_{t-1}, x_t RNN tidak dapat belajar untuk menghubungkan informasi. Penyebabnya adalah memori lama yang tersimpan akan semakin tidak berguna dengan seiringnya waktu berjalan karena tertimpa atau tergantikan dengan memori baru (Bengio dkk., 1994). Berbeda dengan RNN, LSTM tidak memiliki kekurangan tersebut karena LSTM dapat mengatur memori pada setiap masukkannya dengan menggunakan *memory cells* dan unit gerbang. *Long Short-Term Memory* juga memiliki struktur seperti rantai, namun modul pengulangan memiliki struktur yang berbeda. Sebagai gantinya, LSTM memiliki empat lapisan *neural network* tunggal yang berinteraksi dengan cara yang berbeda dengan RNN. Ilustrasi arsitektur LSTM tersebut terdapat pada Gambar 11 sebagai berikut:

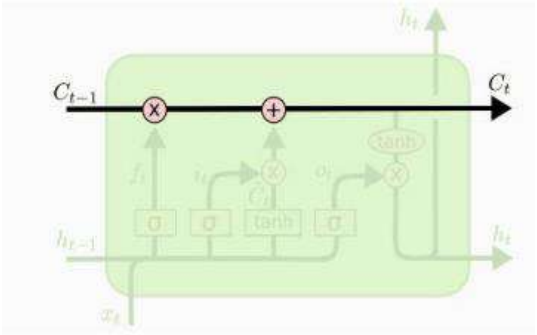


Gambar 11. Modul Pengulangan dalam LSTM berisi empat *layer*
(Alakkari dkk., 2023).

Pada Gambar 11, setiap garis membawa keseluruhan vektor, dari keluaran satu simpul ke masukkan simpul lainnya. Lingkaran berwarna merah muda mewakili operasi *pointwise*, seperti penambahan vektor. Lingkaran berwarna biru merupakan data masukkan, sedangkan kotak berwarna hijau adalah *memory cell*. Kotak kuning adalah lapisan ANN yang dipelajari. Penggabungan garis menunjukkan rangkaian (*concatenation*), sementara garis bercabang menunjukkan bahwa konten disalin dan kemudian salinannya masuk ke lokasi yang berbeda (Le dkk., 2019).

Augenstein dkk. (2016) menyatakan bahwa satu komponen gerbang digunakan saat mengontrol informasi yang masuk ke dalam memori yang bertugas memecahkan masalah *vanishing* dan *exploding gradient*. Koneksi yang berulang menambah keadaan atau memori ke jaringan dan memungkinkanya untuk memanfaatkan pengamatan yang terurut. Memori internal merupakan jaringan keluaran yang bergantung pada konteks terakhir dengan antrian masukkan, tetapi bukan masukkan yang disajikan sebagai jaringan.

Kunci pada algoritma LSTM adalah *cell state* atau garis horisontal yang melewati bagian atas diagram. *Cell state* bekerja seperti *conveyor belt* yang berjalan ke seluruh rantai hanya dengan sedikit interaksi linear. Sangat mudah bagi informasi untuk mengalir begitu saja tanpa perubahan. Gambar 12 merupakan ilustrasi dari *cell state*:

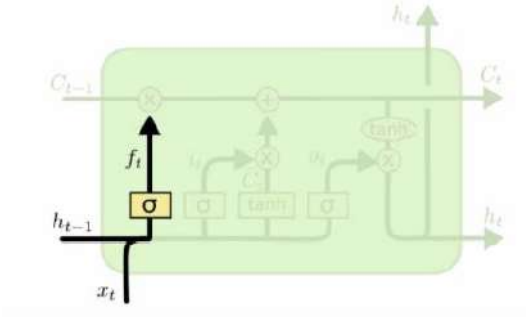


Gambar 12. Ilustrasi *cell state* (Alakkari dkk., 2023).

Long Short-Term Memory memiliki kemampuan untuk menambah dan menghapus informasi ke dalam *cell state* yang diatur secara teliti oleh struktur yang disebut gerbang. Gerbang ini bertugas untuk mengontrol aliran informasi masuk dan keluar

dari *memory cell* yang terdiri dari lapisan *neural network sigmoid* dan operasi perkalian *pointwise*. Berikut ini adalah langkah-langkah untuk mengontrol aliran informasi yang masuk dan keluar pada LSTM:

1. Menentukan informasi yang akan dibuang dari *cell state*. Inti dari tahap ini terletak pada lapisan *sigmoid* atau yang juga disebut gerbang *forget*. Sesuai dengan namanya, lapisan ini berperan untuk menyeleksi informasi yang akan dilupakan. Gerbang *forget* menghasilkan angka 0 dan 1 untuk *cell state* C_{t-1}. Angka 1 merepresentasikan untuk menyimpan nilai memori dan angka 0 merepresentasikan untuk melupakan nilai memori, sehingga gerbang *forget* merupakan gerbang yang memutuskan apakah informasi yang masuk harus dibuang atau tidak dari *cell state*. Perhitungan ini menggunakan data keluaran sebelumnya $h_{(t-1)}$ dan data masukkan x_t dengan alur pada Gambar 13 sebagai berikut:



Gambar 13. Ilustrasi langkah pertama *LSTM*
(Salman dkk., 2021).

Gerbang *forget* diformulasikan dalam Persamaan (5) sebagai berikut:

$$f_t = \sigma\left(W_f * \left[h_{(t-1)}, x_t\right]\right) + b_f \tag{5}$$

dengan:

f_t : gerbang *forget* orde ke *t*
σ : fungsi *sigmoid*
W_f : nilai bobot untuk gerbang *forget*
$h_{(t-1)}$: nilai keluaran sebelum orde ke *t*
x_t : nilai masukkan pada orde ke *t*
b_f : nilai bias pada gerbang *forget*

Nilai bobot dirumuskan dengan Persamaan (6) sebagai berikut:

$$W = \left(-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}\right) \quad (6)$$

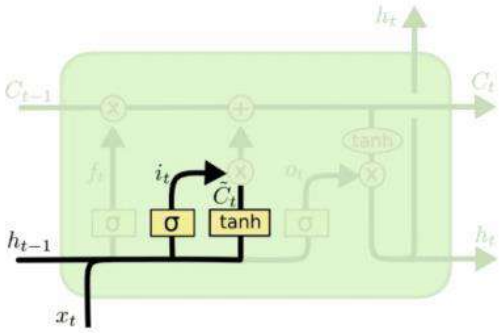
dengan:

W: bobot

d: jumlah data

2. Menentukan informasi yang akan dimasukkan ke dalam *cell state*.

   Pada tahap ini terdapat dua lapisan, yaitu lapisan *sigmoid* dan lapisan $tanh$. Kedua lapisan tersebut dinamakan gerbang *input*. Lapisan *sigmoid* berperan memutuskan nilai mana yang akan diperbarui, sedangkan lapisan $tanh$ berperan untuk membentuk vektor dari nilai kandidat baru $\tilde{C}_t$ yang dapat ditambahkan ke dalam *cell state*. Kemudian *cell state* yang lama, C_{t-1} diperbarui menjadi *cell state* baru, C_t. Alur gerbang *input* ditampilkan pada Gambar 14 sebagai berikut:



Gambar 14. Ilustrasi langkah kedua LSTM
(Salman dkk., 2021).

Gerbang *input* dirumuskan dengan Persamaan (7) sebagai berikut:

$$i_t = \sigma\left(W_i * \left[h_{(t-1)}, x_t\right]\right) + b_i \quad (7)$$

dengan:

i_t : gerbang *input* orde ke *t*

σ : fungsi *sigmoid*

W_i : nilai bobot untuk gerbang *input*

$h_{(t-1)}$: nilai keluaran sebelum orde ke *t*

x_t : nilai masukkan pada orde ke *t*

b_i : nilai bias pada gerbang *input*

Kandidat baru diformulasikan dengan Persamaan (8) sebagai berikut:

$$\tilde{C}_t = tanh\left(W_C * \left[h_{(t-1)}, x_t\right]\right) + b_C \quad (8)$$

dengan:

$\tilde{C}_t$: nilai baru yang ditambahkan ke *cell state* orde ke *t*

$tanh(.)$: fungsi *tanh*(.)

W_C : nilai bobot untuk *cell state*

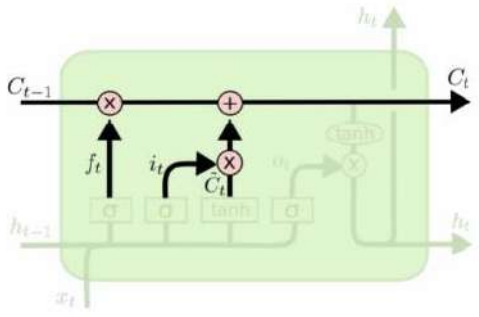
$h_{(t-1)}$: nilai keluaran sebelum orde ke *t*

x_t : nilai masukkan pada orde ke *t*

b_c : nilai bias pada *cell state*

3. Menambahkan informasi baru

Selanjutnya *cell state* yang lama akan diperbaharui menjadi *cell state* yang baru dengan mengalikan *cell state* lama dengan gerbang *forget* untuk menghapus informasi yang telah ditentukan pada lapisan gerbang *forget* kemudian ditambahkan dengan perkalian gerbang *input* dan nilai kandidat baru untuk memperbarui informasi atau memori baru dengan alur pada Gambar 15 sebagai berikut:



Gambar 15. Ilustrasi langkah ketiga *LSTM*
(Salman dkk., 2021).

Adapun rumus penambahan informasi atau memori baru menggunakan Persamaan (9) sebagai berikut:

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \quad (9)$$

dengan:

C_t : *cell state* orde ke *t*

f_t : gerbang *forget* orde ke *t*

C_{t-1} : *cell state* sebelum orde ke *t*

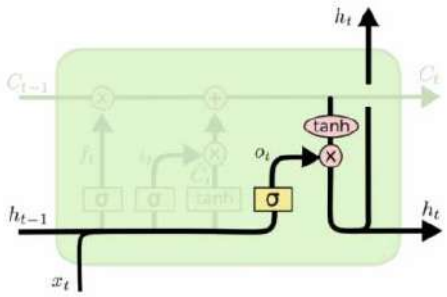
i_t : gerbang *input* orde ke *t*

x_t : nilai masukkan pada orde ke *t*

$\tilde{C}_t$: nilai baru yang ditambahkan ke *cell state* orde ke *t*

4. Menemukan Keluaran

Pada tahap ini merupakan tahapan gerbang *output* yang terdiri dari dua lapisan, yaitu lapisan *sigmoid* dan lapisan *tanh*(.). Lapisan *sigmoid* di eksekusi untuk menentukan bagian dari *cell state* mana yang akan dijadikan keluaran berdasarkan masukkan dan *memory cell*. Keluaran yang dihasilkan harus sesuai dengan *cell state* yang telah diproses sebelumnya sesuai alur pada Gambar 16 berikut:



Gambar 16. Ilustrasi langkah keempat *LSTM*
(Salman dkk., 2021).

Formula gerbang *output* diuraikan dalam Persamaan (10) sebagai berikut:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (10)$$

dengan:

o_t : gerbang *output* orde ke *t*

σ : fungsi *sigmoid*

W_o : nilai bobot untuk gerbang *output*

$h_{(t-1)}$: nilai keluaran sebelum orde ke *t*

x_t : nilai masukkan pada orde ke *t*

b_0 : nilai bias pada gerbang *output*

Kemudian masukkan *cell state* melalui lapisan $tanh$ dikalikan dengan hasil yang diperoleh dari lapisan *sigmoid* agar diperoleh keluaran sesuai dengan keputusan sebelumnya dengan Persamaan (11) sebagai berikut:

$$h_t = o_t * tanh(C_t) \tag{11}$$

dengan:

h_t : nilai keluaran pada orde ke *t*

o_t : gerbang *output* orde ke *t*

$tanh(.)$: fungsi *tanh*(.)

C_t : *cell state* orde ke *t*

Long Short-Term Memory memiliki fungsi untuk mengkomputasi *state* lapisan tersembunyi yang dapat merekam *long-term dependencies* (ketergantungan jangka panjang). Selain itu, LSTM adalah arsitektur jaringan saraf yang cukup baik untuk memproses data sekuensial. Selama proses klasifikasi, dibutuhkan proses *forward propagation* terlebih dahulu agar fungsi aktivasi *softmax* dapat dilakukan.

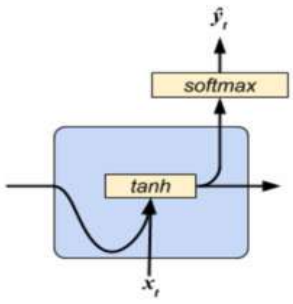
Untuk setiap langkah waktu t, pertama harus mengkalkulasi *state* h_t dari masukkan (x_t) dan *state* sebelumnya (h_{t-1}), masing-masing dikalikan dengan parameter *U* dan W lalu diproses dengan fungsi aktivasi *tanh*(.), seperti pada Persamaan (12) berikut:

$$h_t = tanh\,(\,U \cdot x_t + W \cdot h_{t-1}\,) \tag{12}$$

Selanjutnya dilakukan perhitungan keluaran $\widehat{y_t}$ yang dilakukan dengan cara mengalikan h_t dengan parameter *V* melalui fungsi aktivasi *softmax* pada Persamaan (13) sebagai berikut:

$$\widehat{y_t} = softmax(\,V \cdot h_t\,) \tag{13}$$

Proses tersebut divisualisasikan pada Gambar 17 sebagai berikut:



Gambar 17. Visualisasi Fungsi Aktivasi *Softmax*
(Firmansyah dkk., 2020).

Keterangan:

$tanh$: Fungsi aktivasi tangen hiperbolik dengan rentang (- 1,1)

x_t : Masukkan orde ke *t*

W : Bobot

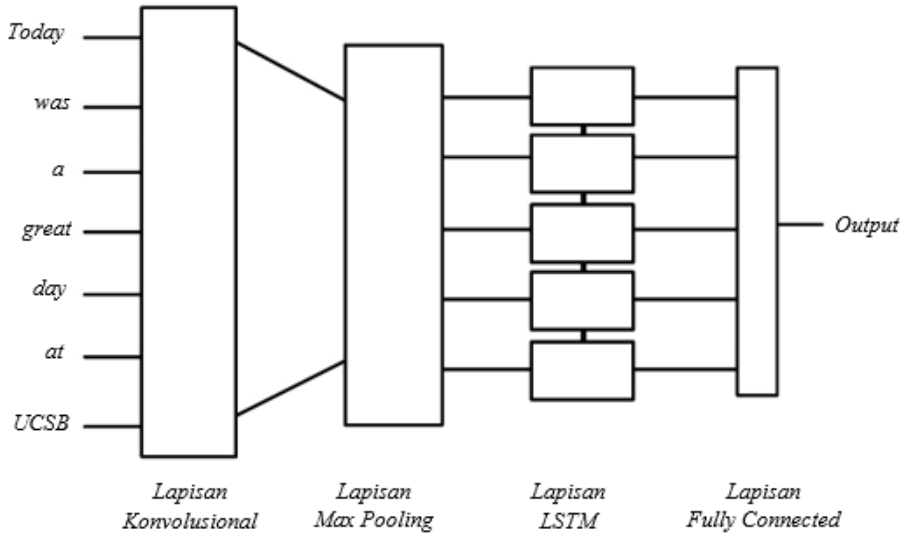
$\widehat{y_t}$: Keluaran orde ke *t*

2.19 *Hybrid Convolution Neural Network - Long Short-Term Memory*

Hybrid Convolution Neural Network - Long Short-Term Memory (*hybrid* CNN – LSTM) adalah kombinasi antara algoritma CNN dan algoritma LSTM. Sesuai dengan urutan namanya, konstruksi model *hybrid* CNN – LSTM meletakkan algoritma LSTM pada lapisan terakhir setelah model jaringan CNN. Cara kerja model *hybrid* CNN – LSTM, yaitu dengan memasukkan vektor masukkan ke dalam proses algoritma CNN terlebih dahulu. Selanjutnya memproses keluaran yang diperoleh dari jaringan CNN dengan mekanisme tiga gerbang yang terdapat pada jaringan LSTM.

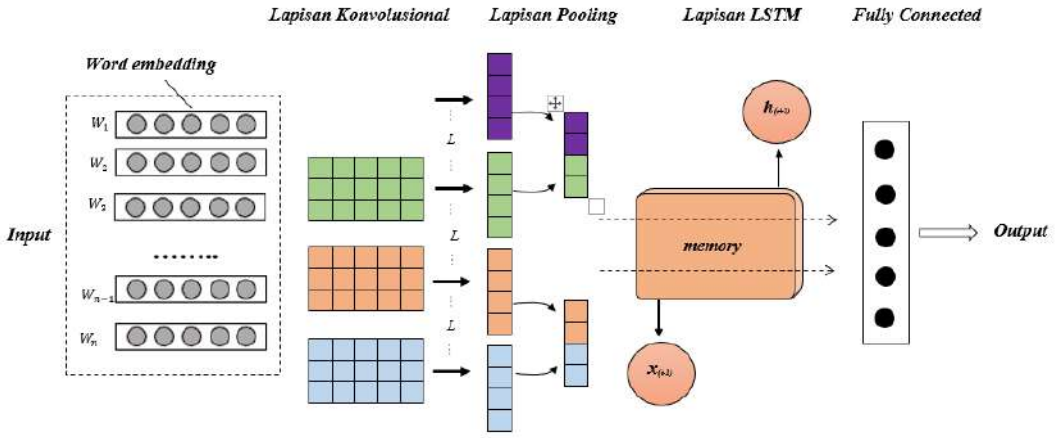
Arsitektur model *hybrid* CNN – LSTM diuraikan pada Gambar 18. Arsitektur model tersebut terdiri dari lapisan konvolusi yang akan menerima masukkan dari *word embedding* sebagai vektor representasi. Kemudian keluaran yang diperoleh akan dikumpulkan ke dimensi yang lebih kecil pada lapisan *pooling* untuk diteruskan sebagai masukkan lapisan LSTM. Keluaran yang diperoleh dari lapisan

LSTM berupa vektor kata dimasukkan ke dalam lapisan *fully-connected*. Setelah itu diperoleh hasil berupa kelas klasifikasi teks. Ide dasar pada model ini adalah jaringan CNN melalui lapisan konvolusi akan mengekstrak fitur lokal, sedangkan jaringan LSTM digunakan untuk mempelajari urutan teks berdasarkan ekstraksi fitur lokal yang diperoleh dari jaringan CNN.



Gambar 18. Model *Hybrid* CNN – LSTM (Mehndiratta dan Soni, 2019).

Zhou dkk. (2015) mengusulkan model baru yang mereka sebut C – LSTM dengan menggabungkan model CNN dan LSTM (*hybrid* CNN – LSTM) untuk representasi kalimat dan klasifikasi teks. Model ini menggunakan CNN untuk mengekstrak rangkaian representasi kata tingkat tinggi, dan dimasukkan ke dalam jaringan LSTM untuk mendapatkan representasi kalimat. Hasil eksperimen menunjukkan bahwa C – LSTM memiliki performa yang lebih baik daripada CNN dan LSTM dalam tugas klasifikasi teks karena mampu menangkap fitur lokal dari kata serta semantik kalimat secara menyeluruh. Gambar 19 merupakan arsitektur model *hybrid* CNN – LSTM yang dapat digunakan untuk klasifikasi teks:

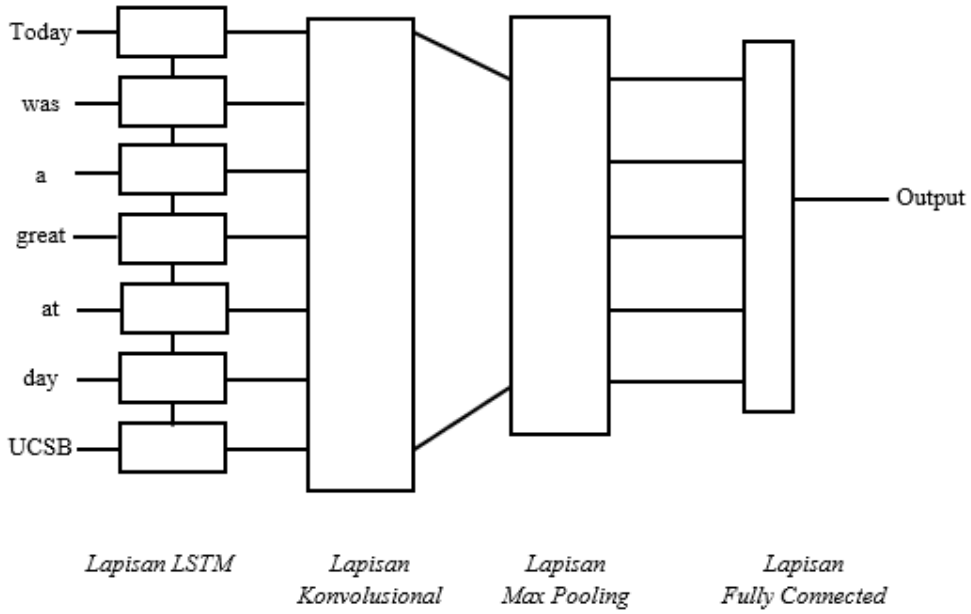


Gambar 19. Arsitektur *Hybrid* CNN – LSTM untuk Klasifikasi Teks.

2.20 *Hybrid Long Short-Term Memory - Convolution Neural Network*

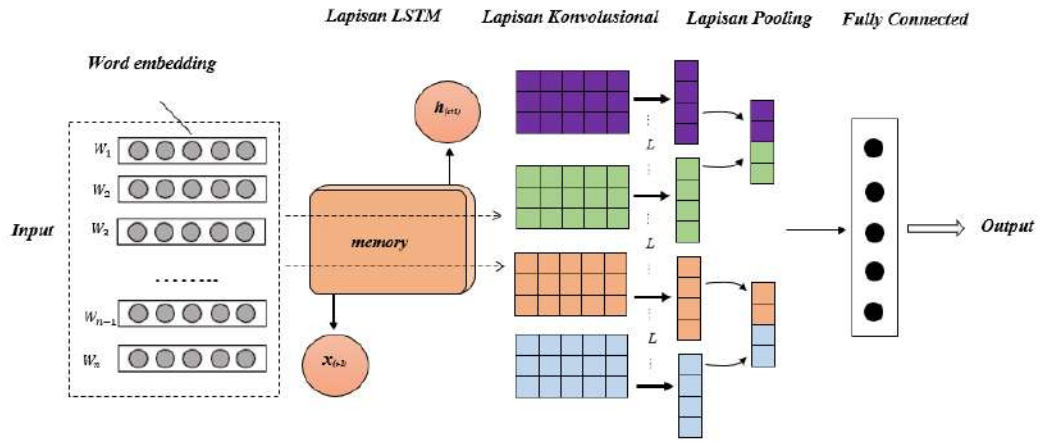
Hybrid Long Short-Term Memory - Convolution Neural Network (*hybrid* LSTM – CNN) adalah kombinasi antara algoritma LSTM dan algoritma CNN. Sesuai dengan urutan namanya, konstruksi model *hybrid* LSTM – CNN meletakkan algoritma CNN pada lapisan terakhir setelah model jaringan LSTM. Cara kerja model *hybrid* LSTM – CNN, yaitu dengan memasukkan vektor masukkan ke dalam algoritma LSTM terlebih dahulu. Selanjutnya keluaran yang diperoleh dari jaringan LSTM akan diproses dengan mekanisme algoritma (tiga lapisan berupa lapisan konvolusi, lapisan *pooling*, dan lapisan *fully connected*) yang terdapat pada jaringan CNN.

Model *hybrid* LSTM – CNN terdiri dari lapisan LSTM yang berfungsi untuk menerima masukkan dari *word embedding* sebagai vektor representasi dan menghasilkan lebih banyak informasi baru. Keluaran dari lapisan LSTM kemudian dimasukkan ke dalam lapisan konvolusi untuk mengekstrak fitur lokal. Terakhir, keluaran lapisan konvolusi akan diubah ke dimensi yang lebih kecil menggunakan lapisan *pooling* dan diteruskan ke lapisan *fully connected* untuk memprediksi hasil akhir kelas. Arsitektur model tersebut divisualisasikan pada Gambar 20 berikut:



Gambar 20. Model *Hybrid* LSTM – CNN (Mehndiratta dan Soni, 2019).

Zhang dkk. (2018) melakukan penelitian klasifikasi teks dengan menggabungkan keunggulan dua model DL, yaitu model LSTM dan model CNN atau dikenal dengan nama *hybrid* LSTM – CNN. Model LSTM digunakan untuk mempertahankan karakteristik informasi yang tersimpan dalam urutan teks yang panjang, sedangkan model CNN digunakan untuk mengekstrak fitur lokal teks. Kinerja model *hybrid* LSTM – CNN lebih baik dibandingkan dengan model CNN maupun model LSTM. Hasil eksperimen menunjukkan bahwa model *hybrid* LSTM– CNN secara efektif dapat meningkatkan akurasi klasifikasi teks. Gambar 21 merupakan arsitektur model *hybrid* LSTM – CNN yang dapat digunakan untuk klasifikasi teks:

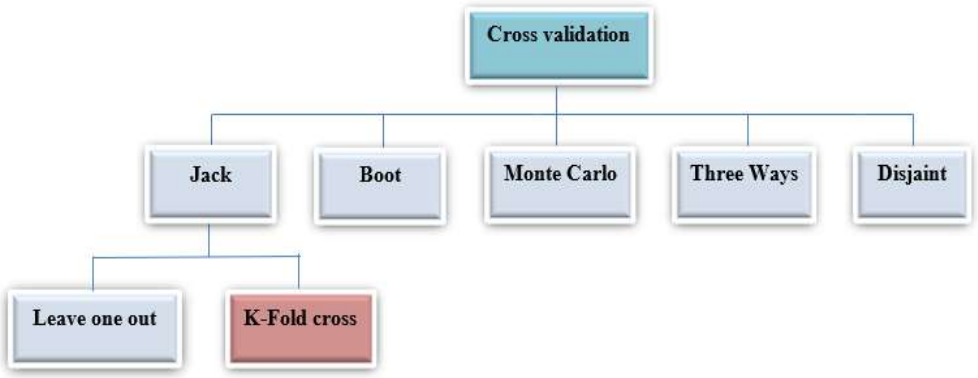


Gambar 21. Arsitektur *Hybrid* LSTM – CNN untuk Klasifikasi Teks.

2.21 K-*fold Cross Validation*

Pada ML maupun DL, model yang sudah dibangun harus melalui proses validasi terlebih dahulu agar model dapat memahami pola data dengan benar sehingga tidak terdapat banyak *noise*. Untuk melakukan validasi model tersebut, perlu dilakukan *cross-validation*.

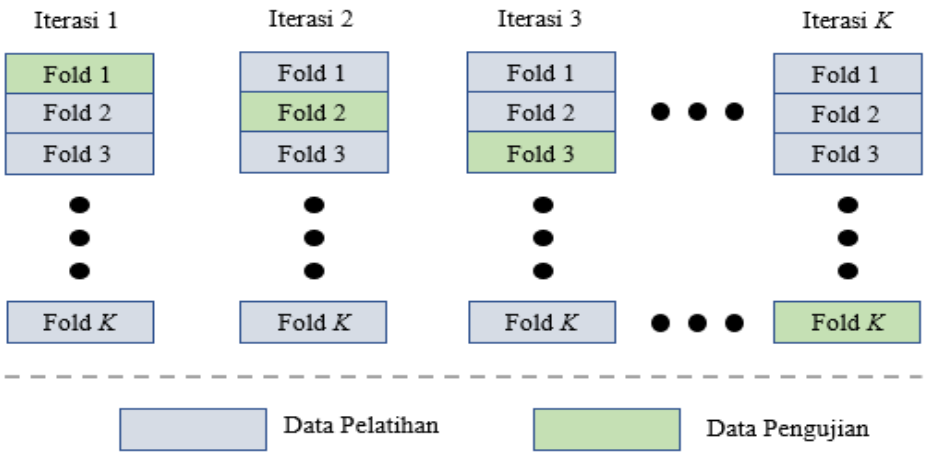
Cross Validation adalah pengukuran penilaian kinerja model prediktif, dan analisis statistik yang akan digeneralisasi ke dalam kumpulan data independen. Terdapat berbagai jenis *Cross Validation* seperti yang dipaparkan pada Gambar 22, misalnya validasi *sub sampling* acak berulang, k-*fold Cross Validation*, *Cross Validation* kx2, *leave one out Cross Validation* dan sebagainya.



Gambar 22. Diagram Tipe Teknik *Cross Validation* (Rohani dkk., 2018).

Menurut Jiang dan Chen (2016), *k-fold Cross Validation* adalah salah satu dari jenis pengujian *Cross Validation* yang berfungsi untuk menilai kinerja proses sebuah metode algoritma dengan membagi data secara acak dan mengelompokkan data tersebut menjadi *k fold*. Kemudian salah satu kelompok *k fold* tersebut akan dijadikan sebagai data pengujian sedangkan sisa kelompok yang lain akan dijadikan sebagai data pelatihan.

Bentuk dasar dari *Cross Validation* adalah k-*fold Cross Validation*. Cara kerja dari k-*fold cross-validation* menurut Bramer (2007) adalah dengan cara membagi data awal secara acak menjadi k *subset* yang saling eksklusif atau *fold*, D_1 sampai D_k, dimana setiap *fold*-nya memiliki ukuran yang sama atau hampir sama. Proses pelatihan dan pengujian akan dilakukan sebanyak *k* kali. Pada perulangan *i*, bagian D_i dipisahkan dan dijadikan sebagai data pengujian dan bagian yang tersisa digunakan secara kolektif untuk melatih model. Artinya, pada perulangan pertama, *subset* D_2, …, Dk bekerja secara kolektif sebagai data pelatihan untuk mendapatkan model, D_1 digunakan sebagai data pengujian. Pada perulangan kedua dari data pelatihan di subset D_1, D_3, …, D_k dan D_2 digunakan sebagai data pengujian dan seterusnya. Tahap selanjutnya semua nilai performa ini akan dicari rata-ratanya dan nilai dengan rata-rata tertinggi akan dipilih sebagai model. Ilustrasi dari k-*fold Cross Validation* terdapat pada Gambar 23 sebagai berikut:



Gambar 23. Ilustrasi *k-fold Cross Validation* (Alsharif dkk., 2021).

Maksud dari ilustrasi di atas adalah, misalnya data D ingin melakukan *Cross Validation* 10 kali *fold*. Artinya data D akan dibagi menjadi 10 subset data yang akan diberi nama D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}. D_1 memiliki lipatan: $fold_2$, $fold_3$, $fold_4$, $fold_5$, $fold_6$, $fold_7$, $fold_8$, $fold_9$, dan $fold_{10}$ sebagai data pelatihan dan $fold_1$ sebagai data data pengujian. D_2 memiliki lipatan: $fold_1$, $fold_3$, $fold_4$, $fold_5$, $fold_4$, $fold_5$, $fold_6$, $fold_7$, $fold_8$, $fold_9$, dan $fold_{10}$ sebagai data pelatihan dan $fold_2$ sebagai data pengujian. Demikian seterusnya hingga D_{10}, sehingga setiap *fold* pernah digunakan sebagai data pengujian sebanyak satu kali.

Han dkk. (2012) juga secara umum merekomendasikan 10-*fold Cross Validation* untuk meningkatkan akurasi, bahkan jika daya komputasi memungkinkan dapat digunakan lebih dari 10-*fold*. Semakin besar *fold* yang digunakan, bias dan varians yang diperoleh relatif lebih rendah. Pada *data mining* dan ML, 10-*fold Cross Validation* adalah yang paling umum digunakan (Rafaeilzadeh dkk., 2009).

K-fold Cross Validation memiliki beberapa kelebihan, diantaranya: dapat mengklasifikasi kumpulan data lebih efisien, membantu dalam mengurangi *overfitting*, dan membantu dalam menghitung nilai optimal *hyperparameter* yang menghasilkan peningkatan efisiensi algoritma. Namun, metode ini juga memiliki kelemahan, yaitu proses komputasi yang digunakan 26 kali lebih besar karena proses yang dilakukan sebanyak *k* kali dan waktu pelatihan yang dibutuhkan meningkat pada saat model melewati beberapa iterasi dari data yang diberikan.

2.22 Evaluasi Model

Proses evaluasi merupakan proses penting dalam mengembangkan model yang baik dengan cara mengetahui performa model tersebut. Performa suatu model dapat diukur melalui beberapa metrik evaluasi, yaitu akurasi, presisi, *recall*, dan *f1-score.* Untuk memperoleh metrik evaluasi tersebut, perlu diketahui beberapa nilai seperti *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Nilai - nilai tersebut nantinya akan ditabulasi secara detail pada suatu matriks

yang disebut *confusion matrix* untuk memvisualisasikan hasil yang diperoleh (Khan dkk., 2023).

2.22.1 *Confusion Matrix Binary*

Pada kasus *binary* (dua kelas), *confusion matrix* (CM) merupakan matriks dengan dimensi 2x2 di mana salah satu label dianggap "positif" dan yang lainya "negatif". Elemen matriks dicirikan berdasarkan label prediksi (positif, negatif), dan hasil perbandingan prediksi dengan label kelas aktual (benar, salah) (Stehman, 1997). Gambar 24 merupakan gambaran sederhana CM pada kasus *binary*:

		Prediksi	
		Negatif **(N)** -	Positif **(P)** +
Aktual	Negatif **-**	True Negative **(TN)**	False Positive **(FP)** **Type I Error**
	Positif +	False Negative **(FN)** **Type II Error**	True Positive **(TP)**

Gambar 24. *Confusion Matrix Binary* (Markoulidakis dkk., 2021).

Nilai TP pada matriks adalah jumlah kelas positif yang dikelompokkan dengan tepat ke dalam kelas positif. Sedangkan jumlah kelas negatif yang dikelompokkan dengan tepat ke dalam kelas negatif disebut nilai TN. Sementara nilai FN adalah jumlah kelas positif yang dikelompokkan ke dalam kelas negatif. Sebaliknya, nilai FP adalah jumlah kelas negatif yang dikelompokkan secara salah ke kelas positif. Apabila nilai TP, TN, FP, dan FN telah diketahui, maka langkah selanjutnya adalah menghitung metrik evaluasi berdasarkan Persamaan pada Tabel 3 sebagai berikut:

Tabel 3. Metrik Performa untuk *Confusion Matrix Binary*

Metrik	**Formula**
Akurasi	$Acc = \frac{TP + TN}{TP + TN + FP + FN}$
Recall (True Positive Rate)	$TPR = \frac{TP}{TP + FP}$
Presisi *(Positive Predictive Value)*	$PPV = \frac{TP}{TP + FN}$
F1-Score	$F1 = 2 \times \frac{TPR.PPV}{TPR + PPV}$

2.22.2 *Confusion Matrix Multiclass*

Pada model *multiclass*, CM yang ditentukan untuk model *binary* tidak sepenuhnya berlaku. *Confusion matrix* pada kasus *multiclass* memiliki dimensi $N \times N$ di mana N adalah jumlah label kelas yang berbeda yang beranggotakan $(C_1, C_2, C_3, \ldots . C_N)$. Oleh karena itu, contoh karakterisasi TP, TN, FP, FN pada kasus *binary* tidak berlaku dalam kasus *multiclass.* Sebaliknya, berdasarkan karakteristik yang diuraikan pada Gambar 25, analisis berfokus pada metrik masing-masing kelas. Selanjutnya, CM untuk model *multiclass* disusun berdasarkan kombinasi yang tepat dari metrik-metrik tersebut (Markoulidakis dkk., 2021).

		Kelas Prediksi			
		C_1	C_2	…	C_N
Kelas Aktual	C_1	$C_{1,1}$	FP	…	$C_{1,N}$
	C_2	FN	TP	…	FN
	…	…	…	…	…
	C_N	$C_{N,1}$	FP	…	$C_{N,N}$

Gambar 25. *Confusion Matrix Multiclass* (Markoulidakis dkk., 2021).

Gambaran umum terkait perhitungan metrik evaluasi pada model *multiclass*, yang mencakup akurasi, *recall*, presisi, dan *f1-score* dijelaskan pada Tabel 4:

Tabel 4. Metrik Performa untuk *Confusion Matrix Multiclass*

Metrik	**Formula**
Akurasi	$Acc = \frac{\sum_{i=1}^{N} TP(C_i)}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}}$
Recall Kelas C_i	$TPR(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)}$
Presisi Kelas C_i	$PPV(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)}$
F1-Score Kelas C_i	$F1(C_i) = 2 \times \frac{TPR(C_i).PPV(C_i)}{TPR(C_i) + PPV(C_i)}$
*Recall (*Rata-rata Makro*)*	$TPR(macro) = \frac{1}{N}\sum_{i=1}^{N} TPR(C_i)$
Presisi *(* Rata-rata Makro *)*	$PPV(macro) = \frac{1}{N}\sum_{i=1}^{N} PPV(C_i)$
F1-Score (Rata-rata Makro *)*	$F1(macro) = 2 \times \frac{TPR(macro).PPV(macro)}{TPR(macro) + PPV(macro)}$

BAB 3
METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian "Pendekatan Baru Ekstraksi Informasi *Biomedical* Big Data *Report* dengan BioWordVec menggunakan Model *Hybrid Long Short-Term Memory – Convolution Neural Network* (LSTM – CNN)" dilakukan pada Tahun Akademik 2022/2023 pada Program Studi Doktor MIPA, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang berada di Jalan Prof. Soemantri Brojonegoro No. 1 Gedong Meneng, Rajabasa, Bandar Lampung.

3.2 Alat dan Bahan

3.2.1 Spesifikasi Komputer

Pada penelitian ini, peneliti menggunakan komputer dengan spesifikasi sebagai berikut:

- *Processor* AMD *Ryzen* 5 PRO 4650G *with Radeon Graphics* 3.69 GHz.
- Pengontrol memori *Ultra-fast* DDR4 UDIMM, PC4-25600 3200 MHz 128 GB RAM.
- *Mainboard* M51 MPG X570 *Gaming Plus*
- SSD XPE M.2 2280 PCle NVMe *spectrix* 540G 1 TB
- VGA Nvidia *GeForce* GTX 1660 *Super Ventus* XS OC
- HDD *Seagate Barracuda* 4 TB
- LED AOC 27-inch 2762 IPS 144 Hz *Gaming Monitor*
- Sistem Operasi *Windows* 10 Pro (64-bit).

3.2.2 *Software* Komputer

Pada penelitian ini, *Software Operating System* (OS) yang digunakan adalah Windows 10 dengan bahasa pemrograman *Python* versi 3.7. Aplikasi bahasa pemrograman *Python* yang digunakan, yaitu *Jupyter Notebook* dan *Google Colab Premium*. Beberapa *library* dalam bahasa pemrograman *Python* yang digunakan dideskripsikan pada Tabel 5 sebagai berikut:

Tabel 5. *Library* Python

Library	Versi	Tugas	Pengembang
NumPy	1.19.5	*Library* NumPy digunakan untuk mengolah data-data dengan karakteristik *big data*, yaitu *volume*, *variety*, *velocity*, dan *veracity*. NumPy adalah operasi data yang mencakup penyimpanan data dan manipulasi data. NumPy dapat digunakan untuk manipulasi data, seperti transformasi bentuk, operasi matematika atau aljabar, dan membangkitkan bilangan acak. Untuk bisa menggunakan NumPy, langkah pertama adalah mengimpor *library* tersebut: import numpy Untuk bisa mengksplor konten dan dokumentasi dari *library* NumPy, gunakan perintah: numpy. <TAB> atau numpy?	Travis E. Oliphant, dkk.
pandas	1.2.4	Fitur utama pandas adalah objek DataFrame yang cepat dan efisien untuk manipulasi data dengan pengindeksan terintegrasi. Pandas juga dapat digunakan untuk mengimpor dan mengekspor data dari berbagai format: comma-separated value (CSV), file teks, Microsoft Excel, *database* SQL, dan format HDF5. Untuk bisa menggunakan Pandas, langkah pertama adalah mengimpor *library* tersebut: import pandas Untuk bisa mengksplor konten dan dokumentasi dari *library* Pandas, gunakan perintah: pandas.<TAB> atau pandas?	Brock Mendel, dkk.

Library	Versi	Tugas	Pengembang
matplotlib	-	Matplotlib adalah *library* lengkap untuk membuat visualisasi statis, animasi, dan interaktif dengan Python. Untuk bisa menggunakan Matplotlib, langkah pertama adalah mengimpor *library* tersebut: import matplotlib Untuk bisa mengeksplor konten dan dokumentasi dari *library* Matplotlib, Anda bisa gunakan perintah: matplotlib<TAB> atau matplotlib?	Thomas. Caswell dkk.
seaborn	0.11.1	Seaborn adalah *library* untuk membuat grafik statistik untuk menghasilkan plot yang informatif. dengan Python. Seaborn bertujuan untuk menjadikan visualisasi sebagai bagian utama dari eksplorasi dan pemahaman data. Untuk bisa menggunakan Seaborn, langkah pertama adalah mengimpor *library* tersebut: import seaborn Untuk bisa mengeksplor konten dan dokumentasi dari library Seaborn, gunakan perintah: seaborn.<TAB> atau seaborn?	Michael Waskom
torch	1.10.0	PyTorch adalah paket Python yang menyediakan dua fitur tingkat tinggi, yaitu: - Komputasi tensor (seperti NumPy) dengan akselerasi GPU yang kuat - *Deep Neural Network* yang dibangun di atas sistem autograd berbasis pita Untuk bisa menggunakan PyTorch, langkah pertama adalah mengimpor *library* tersebut: import torch Untuk bisa mengksplor konten dan dokumentasi dari *library* PyTorch, gunakan perintah: torch.<TAB> atau torch?	Eli Eurigas, dkk.

Library	Versi	Tugas	Pengembang
imblearn	0.8.1	Imbalanced-learn (imblearn) adalah *library* python yang menawarkan sejumlah teknik *resampling* yang biasa digunakan pada kumpulan data yang tidak seimbang. Untuk bisa menggunakan Imbalanced-learn, langkah pertama adalah mengimpor *library* tersebut: import imblearn Untuk bisa mengeksplor konten dan dokumentasi dari *library* Imbalanced-learn, gunakan perintah: imblearn.<TAB> atau imblearn?	Guillaume Lema Tre, dkk.
wordcloud	1.8.1	Wordcloud adalah *library* yang digunakan sebagai generator kata (*word*) dalam bentuk *cloud*. Untuk bisa menggunakan wordcloud, langkah pertama adalah mengimpor *library* tersebut: import wordcloud Untuk bisa mengeksplor konten dan dokumentasi dari *library* wordcloud, gunakan perintah: wordcloud.<TAB> atau wordcloud?	Andreas Mueller, dkk.
sklearn	0.24.1	Scikit-Learn adalah *library* python yang menyediakan kakas sederhana dan efisien untuk menerapkan tugas analisis data dan pembelajaran mesin, seperti klasifikasi, regresi, klasterisasi, reduksi dimensi, pemilihan model, dan pemrosesan. Untuk bisa menggunakan Scikit-Learn, langkah pertama adalah mengimpor *library* tersebut: import sklearn Untuk bisa mengeksplor konten dan dokumentasi dari *library* Scikit-Learn, gunakan perintah: sklearn<TAB> atau sklearn?	Andreas Mueller, dkk.

Library	Versi	Tugas	Pengembang
tensorflow	2.3.0	TensorFlow adalah *library* perangkat lunak *opensource* untuk komputasi numerik yang memiliki performa tinggi. Arsitekturnya yang fleksibel memungkinkan penerapan komputasi yang mudah di berbagai platform (CPU, GPU, TPU). Untuk bisa menggunakan TensorFlow, langkah pertama adalah mengimpor *library* tersebut: import tensorflow Untuk bisa mengeksplor konten dan dokumentasi dari *library* TensorFlow, gunakan perintah: tensorflow.<TAB> atau tensorflow?	Austin Anderson, dkk.
nltk	3.6.1	NLTK adalah rangkaian modul python *open source*, yang berisi kumpulan data, dan tutorial yang mendukung penelitian dan pengembangan di *Natural Language Processing*. Untuk bisa menggunakan NLTK, langkah pertama adalah mengimpor *library* tersebut: import nltk Untuk bisa mengeksplor konten dan dokumentasi dari *library* NLTK, gunakan perintah: nltk.<TAB> atau nltk?	Ilia Kurenkov, dkk.
gensim	4.1.2	Gensim adalah *library* python untuk *topic modelling*, *document indexing*, dan *similarity retrieval* dengan korpora besar. Untuk bisa menggunakan Gensim, langkah pertama adalah mengimpor *library* tersebut: import gensim Untuk bisa mengeksplor konten dan dokumentasi dari *library* Gensim, gunakan perintah: gensim.<TAB> atau gensim?	Michael Penkov, dkk.

3.3 Data Penelitian

Data yang digunakan dalam penelitian ini adalah GENIA *Biomedical Event*, yaitu data yang berisi teks biomedis. Kumpulan data ini memaparkan tentang efek obat terhadap seseorang, serta juga dapat digunakan untuk mengidentifikasi kondisi medis tertentu pada seseorang. Data tersebut diperoleh dari kaggle.com, dan terdiri dari 3 kumpulan data: kumpulan data pelatihan (8,000+ *sentences*), kumpulan data pengujian (3,000 *sentences*), dan kumpulan data validasi (3,000 *sentences*). Setiap kumpulan data mencakup 4 variabel, antara lain:

1. *Sentence*: Teks biomedis asli
2. *TriggerWord*: Kata pemicu
3. *TriggerWordLoc*: Posisi *trigger word* dalam teks
4. *EventType*: Jenis peristiwa yang berasosiasi dengan *trigger word*.

Sumber asli kumpulan data berasal dari BioNLP *Shared Task* 2011 (Dec 2008-2011) yang diperoleh dari:

- PubMed: *Database* Perpustakaan Kedokteran Nasional AS
- *Open Access Subset of PubMed Central* (PMC), yaitu arsip digital gratis untuk jurnal *biomedical* dan *life science* di Institut Nasional AS

Kumpulan data yang digunakan pada penelitian ini adalah data pelatihan, yang berisi lebih dari 8,000 *sentences*. Data ini kemudian di *preprocessing* sesuai dengan kebutuhan tugasnya. Namun, karena datanya tidak memiliki label, maka perlu dilakukan proses pelabelan. Prosedur pelabelan yang digunakan yaitu dengan metode *Cosine Similarity* dan *Word Mover's Distance*. Selain itu, distribusi datanya tidak seimbang, sehingga perlu dilakukan *resampling* terlebih dahulu. Metode *resampling* yang dipakai, yaitu gabungan (*hybrid*) antara *Random Undersampling* dan *Random Oversampling.* Langkah ini diperlukan karena sebagian besar metode ML tidak *robust* terhadap data pelatihan yang tidak seimbang.

3.4 Tahapan Penelitian

Penelitian ini dilakukan dalam beberapa tahap sebagai berikut:

3.4.1 Tahap Pendahuluan

Pada tahap ini dilakukan pencarian literatur yang berkaitan dengan topik "Pendekatan Baru Ekstraksi Informasi *Biomedical* Big Data *Report* Dengan Biowordvec Menggunakan Model *Hybrid Long Short-Term Memory – Convolution Neural Network* (LSTM – CNN)". Proses pencarian literatur ini dibarengi dengan proses pemahaman dan pemetaan penelitian. Hasil pekerjaan pada tahap pendahuluan dirangkum dan dibuat menjadi proposal penelitian.

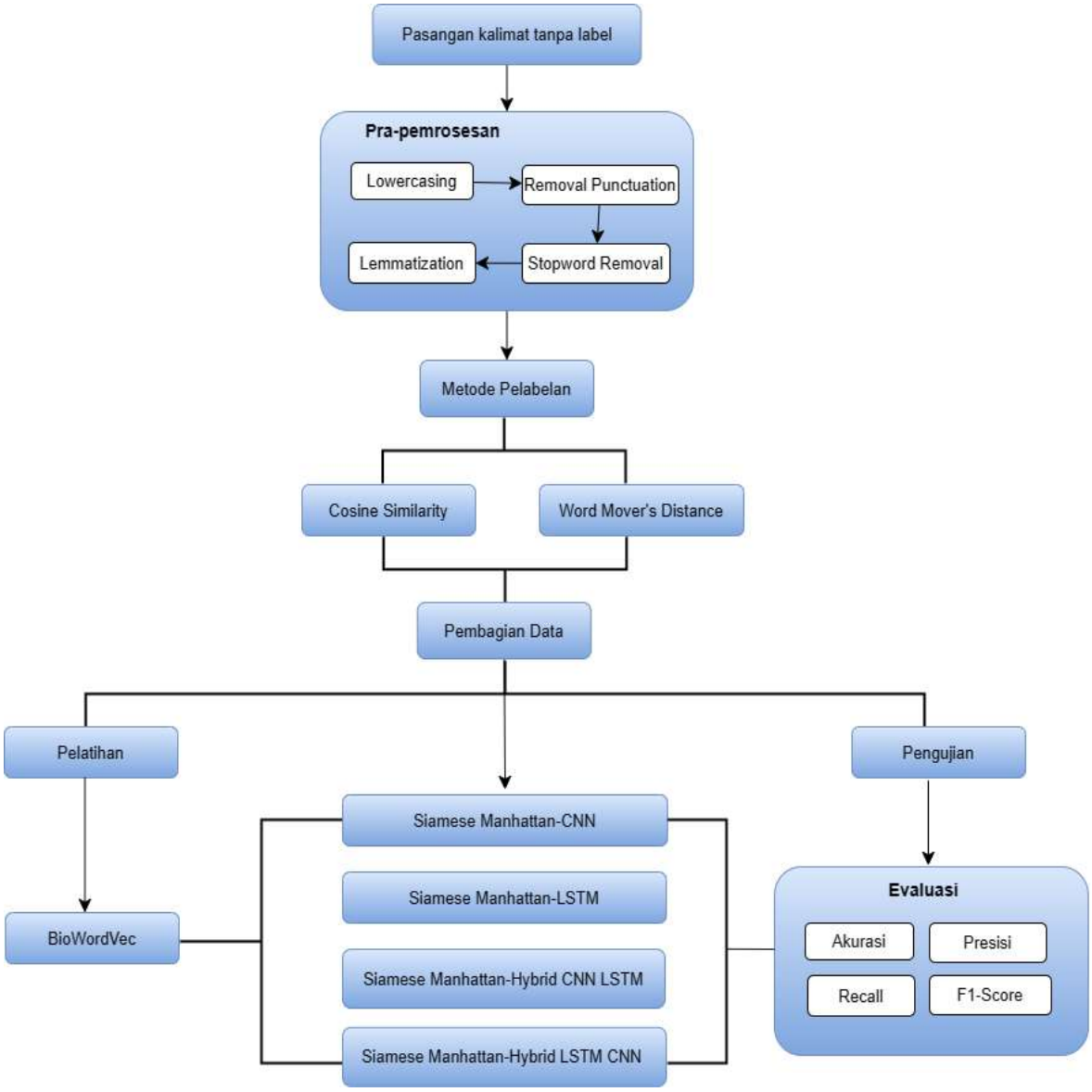
3.4.2 Tahap Penelitian

Pada tahap ini dilakukan kajian ekstraksi informasi. Kajian ini dibagi menjadi dua tahapan, yaitu tahapan STS dan tahapan klasifikasi teks. Kerangka penelitian untuk kajian ekstraksi informasi divisualisasikan oleh Gambar 26 sebagai berikut:

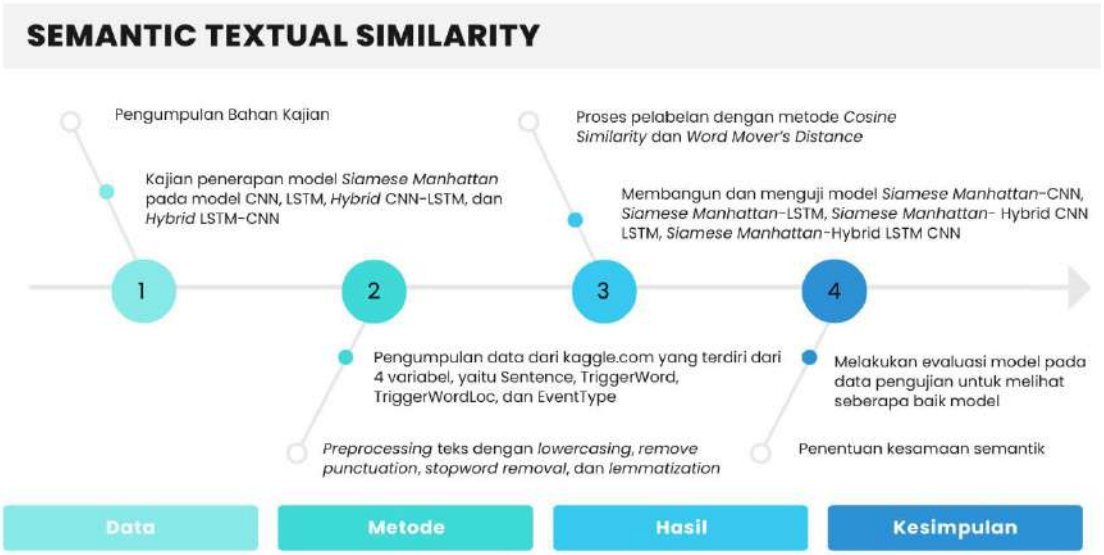
Gambar 26. Kerangka Penelitian untuk Ekstraksi Informasi.

3.4.2.1 Tahap Penelitian I

Langkah-langkah yang dilakukan pada tugas STS dideskripsikan melalui diagram alir dan *fish bone* penelitian pada Gambar 27 dan Gambar 28 sebagai berikut:



Gambar 27. Diagram Alir Penelitian STS.



Gambar 28. *Fish Bone* Penelitian STS.

1. Pengumpulan Data

   Pada tahap ini dilakukan proses pengumpulan data dari kaggle.com dan terdapat tiga kumpulan data yang disimpan dalam *file* berekstensi csv. Tiga kumpulan data tersebut berupa data pelatihan dengan jumlah data sebanyak 8,666 *sentences*, data validasi dengan jumlah data sebanyak 2,866 *sentences*, dan data pengujian dengan jumlah data sebanyak 3,360 *sentences*. Masing-masing data terdiri dari 3 variabel, yaitu: *Sentence*, *Trigger Word*, dan *EventType*. Masing-masing variabel akan dihitung kesamaan semantiknya dengan variabel lain dengan ketentuan sebagai berikut: *Sentence-TriggerWord*, *Sentence-EventType*, dan *TriggerWord-EventType*. Namun variabel *Trigger Word*, dan *EventType* pada data validasi terdapat banyak data yang kosong. Demikian juga pada data pengujian secara keseluruhan data kosong. Sehingga data yang digunakan dari proses pengumpulan data tersebut adalah data pelatihan. Data pelatihan tersebut di-*upload* ke *Google Drive*, kemudian dilakukan *import* data dan selanjutnya disimpan ke dalam pemrograman *Python* dengan menggunakan aplikasi *Google Colabs* dan *Jupyter Notebook*. Selain dari data, pendekatan yang dilakukan untuk memperoleh informasi

adalah melalui pendekatan studi pustaka. Studi pustaka dilakukan dengan cara mempelajari dan memahami buku-buku, jurnal-jurnal, dan beberapa artikel yang berhubungan dengan topik penelitian yang dibahas.

2. *Preprocessing* dan analisis data

   Melakukan *preprocessing* pada data teks dengan *library* "nltk" yang meliputi:

   a. *Lowercasing;* merupakan proses mengubah huruf kapital menjadi huruf kecil. Proses ini penting karena variasi dalam kapitalisasi, misalnya Leukemia versus leukemia dibaca oleh program komputer sebagai vektor kata yang berbeda sehingga dapat memberikan hasil yang berbeda.
   b. *Remove Punctuation*; tanda baca seperti titik, koma, tanda seru, tanda tanya, dan karakter dihapus karena program komputer tidak dapat mengerti penggunaan tanda baca sehingga keberadaan tanda baca dalam teks dianggap sebagai *noise*.
   c. *Stopword Removal*; kata-kata yang tidak memiliki arti, kata-kata khusus dalam domain biomedis, dan yang memiliki frekuensi tinggi dalam teks dihapus. Selain itu, kata yang jarang juga dilakukan penghapusan karena jumlahnya terlalu banyak dan tidak membantu dalam tugas STS.
   d. *Lemmatization*; proses ini merupakan tahapan penting dalam teks *preprocessing*. *Lemmatization* adalah sebuah proses di mana kata diubah ke dalam bentuk morfologi dasar (*lemma*).

3. *Labelling*

   Setelah tahapan *preprocessing* terhadap data dilakukan, langkah selanjutnya adalah melakukan tahapan pelabelan pada *biomedical* big data *report* dengan menggunakan metode CS dan WMD.

4. Ekstraksi Fitur

   Tahap ekstraksi fitur dilakukan setelah tahap pelabelan. Ekstraksi fitur dilakukan untuk memperoleh atribut atau kualitas suatu kata. Pada tahap ini, setiap kata diubah menjadi representasi vektor yang menunjukkan polaritas kata tersebut. Teknik yang digunakan melibatkan penggunaan model *pre-trained word embedding* yang dikenal sebagai BioWordVec. Representasi vektor kata ini melibatkan konversi sekumpulan kata menjadi vektor.

Prosesnya meliputi tokenisasi, yaitu pemisahan kata dari data teks, dan pengurutan sekuensial, yaitu menyusun kata dari data teks yang diberi token. Selain itu, *padding* digunakan untuk memastikan bahwa panjang kata dalam data teks disamakan berdasarkan jumlah kata maksimal dalam setiap kalimat.

5. Membangun dan menguji model

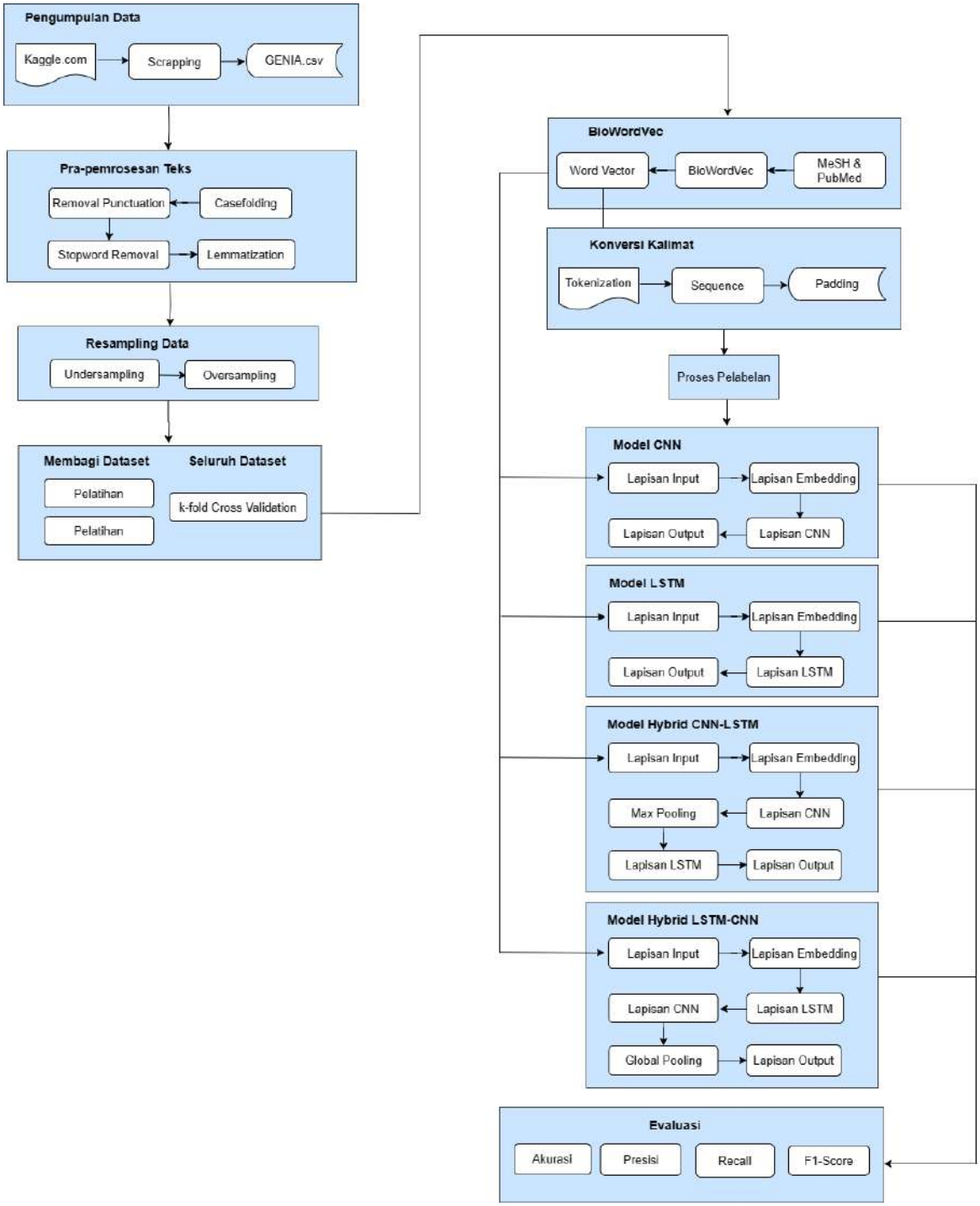
   Tahapan yang dilakukan saat ini adalah implementasi proses penentuan STS yang telah melalui ekstraksi fitur menggunakan model *Siamese Manhattan* - CNN, *Siamese Manhattan* - LSTM, *Siamese Manhattan - hybrid* CNN– LSTM, dan *Siamase Manhattan - hybrid* LSTM–CNN. Keempat model tersebut dibangun dengan menggunakan *library* "tensorflow.keras". Model *Siamese Manhattan* - CNN menerapkan dua tahapan, yaitu proses konvolusi dan metode *pooling*. Model *Siamese Manhattan* - LSTM menggunakan pendekatan sekuensial untuk mengurai vektor fitur teks. Model *Siamese Manhattan - hybrid* CNN–LSTM terdiri dari dua tahap utama: tahap CNN dan tahap LSTM. Tahapan CNN terdiri dari dua tahap yaitu proses konvolusi dan proses *pooling*. Hasil CNN dijadikan sebagai masukan untuk model LSTM. Sedangkan model *Siamese Manhattan - hybrid* LSTM–CNN terdiri dari dua tahap utama: tahap LSTM dan tahap CNN. Selama tahap LSTM, tahapan pemrosesan berurutan digunakan untuk mengubah masukan menjadi vektor fitur teks. Hasil yang diperoleh dari model LSTM digunakan sebagai masukkan pada CNN. Performa keempat model DL dievaluasi dengan membandingkan metode pelabelan dan jumlah lapisan tersembunyi yang digunakan selama fase pelatihan.

6. Evaluasi model.

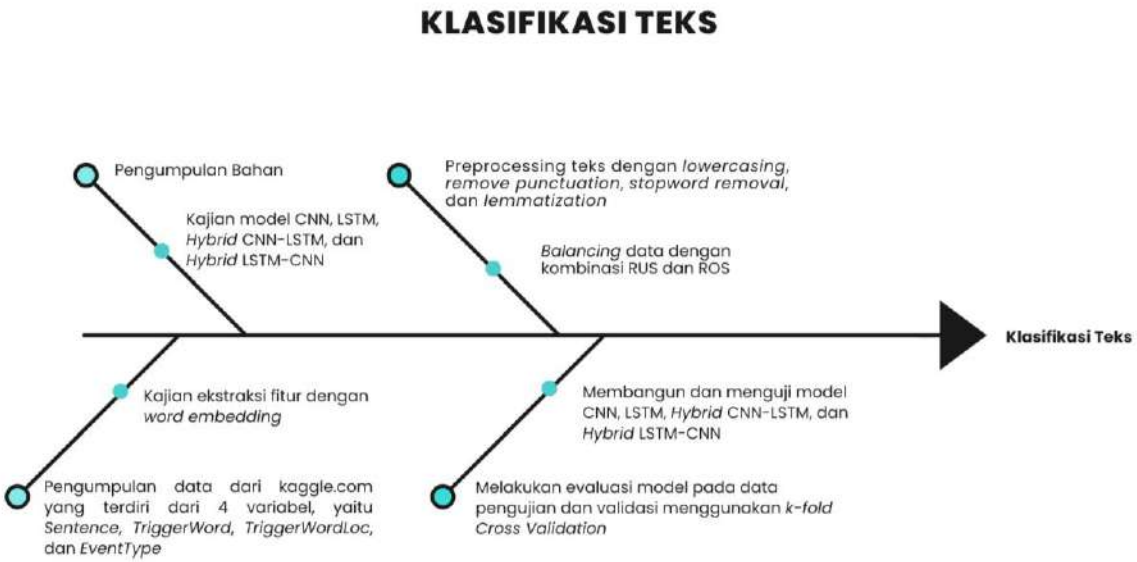
   Setelah dilakukannya proses pelatihan, model yang sudah dibangun akan dilakukan pengujian dengan menggunakan data pengujian untuk mengevaluasi dan melihat seberapa baik performa model yang telah dibangun dengan menggunakan *confusion matrix*. Berdasarkan metrik pengukuran tersebut, nilai akurasi, presisi, *recall*, dan *f1-score* dapat dihitung. Hasil perhitungan STS yang benar berada tepat pada garis diagonal matriks. Sementara distribusi kesalahan perhitungan STS berada di luar garis diagonal.

3.4.2.2 Tahap Penelitian II

Langkah-langkah yang dilakukan pada tugas klasifikasi teks diuraikan melalui diagram alir dan *fish bone* penelitian pada Gambar 29 dan Gambar 30 sebagai berikut:



Gambar 29. Diagram Alir Penelitian Klasifikasi Teks.



Gambar 30. *Fish Bone* Penelitian Klasifikasi Teks.

1. Pengumpulan Data

   Pada tahap ini dilakukan proses pengumpulan data dari kaggle.com dan terdapat tiga kumpulan data yang disimpan dalam *file* berekstensi csv. Tiga kumpulan data tersebut berupa data pelatihan dengan jumlah data sebanyak 8,666 *sentences*, data validasi dengan jumlah data sebanyak 2,866 *sentences*, dan data pengujian dengan jumlah data sebanyak 3,360 *sentences*. Masing-masing data terdiri dari 4 variabel, yaitu: *Sentence*, *Trigger* Word, T*riggerWordLoc*, dan *EventType*. Namun variabel *Trigger Word*, T*riggerWordLoc*, dan *EventType* pada data validasi terdapat banyak data yang kosong. Sedangkan pada data pengujian secara keseluruhan data kosong. Sehingga data yang digunakan dari proses pengumpulan data tersebut adalah data pelatihan. Data pelatihan tersebut di-*upload* ke *Google Drive*, kemudian dilakukan *import* data dan selanjutnya disimpan ke dalam pemrograman *Python* dengan menggunakan aplikasi *Google Colabs* dan *Jupyter Notebook*. Selain dari data, pendekatan yang dilakukan untuk memperoleh informasi adalah melalui pendekatan studi pustaka. Studi pustaka dilakukan dengan teknik mempelajari dan memahami buku-buku, jurnal-jurnal, dan beberapa artikel yang berhubungan dengan topik penelitian yang dibahas.

2. *Preprocessing* dan analisis data

   Melakukan *preprocessing* pada data teks dengan *library* "nltk" yang meliputi:

   a. *Case Folding*, yaitu mengubah kata-kata yang mengandung huruf kapital menjadi huruf kecil.
   b. *Stopwords Removal*, yaitu menghilangkan kata-kata yang tidak memiliki makna.
   c. *Punctuation Marks Removal*, yaitu menghilangkan tanda-tanda baca.
   d. Menghilangkan karakter yang muncul berlebihan pada suatu kata.
   e. Menghilangkan karakter yang terlalu sering muncul dan terlalu jarang muncul.
   f. *Lemmatization*, yaitu mengembalikan suatu kata menjadi kata dasar atau formalisasi kata.

   Setelah dilakukan *preprocessing* data, juga dilakukan analisis eksplorasi data dengan visualisasi menggunakan diagram batang dan *Wordcloud* dengan tujuan untuk mendapatkan gambaran penyebaran data. Berdasarkan gambaran data tersebut akan dilakukan keseimbangan data jika kelas klasifikasi tidak seimbang dengan menggunakan *resampling* data baik dengan *Random Oversampling*, *Random Undersampling* maupun kombinasi antara keduanya.

3. Ekstraksi Fitur

   Tahap ekstraksi fitur dilakukan setelah tahap pelabelan. Ekstraksi fitur dilakukan untuk memperoleh atribut atau kualitas suatu kata. Pada tahap ini, setiap kata diubah menjadi representasi vektor yang menunjukkan polaritas kata tersebut. Teknik yang digunakan melibatkan penggunaan model *pre-trained word embedding* yang dikenal sebagai BioWordVec. Representasi vektor kata ini melibatkan konversi sekumpulan kata menjadi vektor. Prosesnya meliputi tokenisasi, yaitu pemisahan kata dari data teks, dan pengurutan sekuensial, yaitu menyusun kata dari data teks yang diberi token. Selain itu, *padding* digunakan untuk memastikan bahwa panjang kata dalam data teks disamakan berdasarkan jumlah kata maksimal dalam setiap kalimat.

4. Membangun dan menguji model

   Pada tahap ini dilakukan proses klasifikasi data setelah melalui proses ekstraksi fitur menggunakan berbagai model DL antara lain: CNN, LSTM, *hybrid* CNN – LSTM, dan *hybrid* LSTM – CNN. Keempat model DL tersebut dibangun dengan menggunakan *library* "tensorflow.keras". Model CNN terdiri dari dua tahap: proses konvolusi dan prosedur *pooling*. Model LSTM menggunakan pendekatan sekuensial untuk memproses vektor fitur yang mewakili teks. Model *hybrid* CNN – LSTM terdiri dari dua tahap utama: tahap CNN dan tahap LSTM. Selama tahap CNN, dua proses digunakan: konvolusi dan pengumpulan. Hasil CNN dijadikan sebagai masukan untuk model LSTM. Sedangkan model *hybrid* LSTM – CNN terdiri dari dua tahap utama: tahap LSTM dan tahap CNN. Selama tahap LSTM, tahapan pemrosesan berurutan digunakan untuk mengubah masukan menjadi vektor fitur teks. Keluaran model LSTM digunakan sebagai masukan model CNN. Selama pengembangan keempat model DL ini, berbagai skenario dihasilkan dengan menyesuaikan berbagai estimasi parameter, termasuk jumlah lapisan konvolusi, jumlah *epoch*, ukuran *batch*, *optimizer*, dan *learning rate*. Performa keempat model DL dievaluasi dengan dua cara: dengan mempartisi data menjadi set pelatihan dan pengujian, dan dengan melakukan validasi data menggunakan *k-fold Cross-Validation*.

5. Evaluasi model.

   Setelah dilakukannya proses pelatihan, model DL yang sudah dibangun dilakukan pengujian pada data pengujian untuk mengevaluasi dan menetapkan seberapa baik performa dari model DL yang telah dibangun dengan menggunakan *confusion matrix*. Pengujian terhadap data pengujian juga digunakan untuk mengukur sejauh mana model DL yang dihasilkan mampu melakukan klasifikasi teks dengan tepat.

BAB 6
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan evaluasi, analisis, dan perbandingan yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut:

1. *Word embedding* khusus biomedis, yaitu BioWordVec diterapkan sebagai lapisan *embedding* pada model DL untuk merepresentasikan kata menjadi sebuah vektor *low dense* yang bertujuan untuk menangkap makna semantik dan sintaktik dari kalimat biomedis dengan lebih baik.
2. Label data untuk tugas STS menggunakan metode *Cosine Similarity* diperoleh dengan cara mengukur kosinus sudut antara dua vektor yang diproyeksikan dalam ruang multidimensi. Sementara, metode *Word Mover's Distance* memperoleh label dengan cara menghitung ketidaksamaan antara dua dokumen tekstual sebagai jarak minimum yang diperlukan untuk kata-kata yang ada dalam satu dokumen untuk dipindahkan agar sesuai dengan kata-kata yang ada di dokumen lainnya.
3. Model STS dibangun dengan menerapkan arsitektur *Siamese Manhattan* pada model CNN, LSTM, *hybrid* CNN – LSTM, dan *hybrid* LSTM – CNN. Sehingga model ini dinamakan model *Siamese Manhattan* – CNN, *Siamese Manhattan* – LSTM, *Siamese Manhattan – hybrid* CNN - LSTM, dan *Siamese Manhattan – hybrid* LSTM – CNN. Lebih lanjut, model STS dibangun dan dibandingkan berdasarkan jumlah lapisan tersembunyi yang digunakan untuk menemukan model terbaik dalam menentukan kesamaan semantik teks biomedis. Model tersebut juga dibandingkan berdasarkan metode pelabelan karena data yang digunakan tidak memiliki label. Jumlah lapisan tersembunyi yang digunakan adalah 2 dan 3, sedangkan metode pelabelan yang digunakan adalah metode *Cosine Similarity* dan metode *Word Mover's Distance*.

4. Masalah ketidakseimbangan kelas pada penelitian tahap II ditangani dengan melakukan kombinasi dua prosedur *resampling*. Prosedur pertama, jumlah kelas mayoritas dihapus atau dikurangi secara acak menggunakan metode *Random Undersampling*. Prosedur kedua, kelas minoritas direplikasi secara acak menggunakan metode *Random Oversampling* hingga jumlah data pada kelas mayoritas maupun kelas minoritas seimbang.
5. Model klasifikasi teks dibangun dengan mengaplikasikan skema pembagian data berdasarkan metode *train-test split* dan *k-fold Cross Validation* pada model DL, yaitu CNN, LSTM, *hybrid* CNN-LSTM, dan *hybrid* LSTM-CNN untuk menemukan model terbaik dalam melakukan klasifikasi teks biomedis.
6. Berdasarkan akurasi, presisi, *recall*, dan *f1-score* yang diperoleh pada penelitian tahap I dan tahap II, model terbaik dalam melakukan ekstraksi informasi dari *biomedical* big data *report* adalah model *hybrid* LSTM – CNN. Model tersebut terbukti memiliki performa yang lebih unggul dibandingkan model CNN, LSTM, dan model *hybrid* CNN – LSTM dengan nilai akurasi mencapai 100% pada tugas STS dan mencapai 99% untuk tugas klasifikasi teks.

6.2 Saran

1. Berdasarkan hasil penelitian STS, diperoleh satu kelas sehingga perlu dikaji lebih lanjut untuk kasus *one classification.*
2. Penggalian informasi yang lebih mendalam dapat dilakukan dengan mengidentifikasi dan mendeteksi entitas dari teks biomedis.
3. Kajian selanjutnya dapat dilakukan dengan mengekstraksi hubungan yang terjadi antar entitas biomedis sebagai pengembangan lebih lanjut dari tugas STS.
4. Kajian tugas klasifikasi yang telah dilakukan perlu ditinjau kembali terutama pada label yang berupa label umum seperti *cause, change, high, found, effect,* dan sebagainya. Sebaiknya label-label tersebut dikumpulkan menjadi satu kategori label.
5. Kajian tugas klasifikasi yang telah dilakukan, perlu dilakukan pengembangan dalam metode pembagian data, yaitu dengan menambahkan data validasi. Hal

ini dilakukan untuk menghindari terjadinya bias karena penggunaan data pengujian yang berulang untuk seleksi model dan evaluasi akhir model.

6. Pada penelitian selanjutnya untuk tugas klasifikasi dapat dilakukan pengembangan lebih lanjut menggunakan *multiclass* dengan *multilabel* yang berarti bahwa bukan hanya label pertama yang digunakan pada kelas dalam variabel *TriggerWord* dan variabel *EventType*.

DAFTAR PUSTAKA

Ahmed, N., Yigit, A., Isik, Z., & Alpkocak, A. (2019). Identification of leukemia subtypes from microscopic images using convolutional neural network. *Diagnostics*, *9*(3). https://doi.org/10.3390/diagnostics9030104

Alakkari, K., Subhi, A. A., Alkattan, H., Kadi, A., Malinin, A., Potoroko, I., Abotaleb, M., & El-Kenawy, E. S. M. (2023). Forecasting covid-19 infection using encoder-decoder lstm and attention lstm algorithms. *Journal of Intelligent Systems and Internet of Things*, *8*(2), 20–33. https://doi.org/10.54216/JISIoT.080202

Alam, F., Afzal, M., & Malik, K. M. (2020). Comparative Analysis of Semantic Similarity Techniques for Medical Text. *International Conference on Information Networking*, *2020-Janua*, 106–109. https://doi.org/10.1109/ICOIN48656.2020.9016574

Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, *36*(4), 7764–7772. https://doi.org/10.1016/j.eswa.2008.11.022

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques*. http://arxiv.org/abs/1707.02919

Almazaydeh, L., Abuhelaleh, M., Tawil, A. Al, & Elleithy, K. (2023). Clinical Text Classification with Word Representation Features and Machine Learning Algorithms. *International Journal of Online and Biomedical Engineering*, *19*(4), 65–76. https://doi.org/10.3991/ijoe.v19i04.36099

Almeida, F., & Xexéo, G. (2019). *Word Embeddings: A Survey*. http://arxiv.org/abs/1901.09069

ALRashdi, R., & O'Keefe, S. (2019). *Deep Learning and Word Embeddings for Tweet Classification for Crisis Response*. http://arxiv.org/abs/1903.11024

Alsharif, R., Al-Issa, Y., Alqudah, A. M., Qasmieh, I. A., Mustafa, W. A., & Alquran, H. (2021). Pneumonianet: Automated detection and classification of pediatric pneumonia using chest x-ray images and cnn approach. *Electronics (Switzerland)*, *10*(23). https://doi.org/10.3390/electronics10232949

Alshdaifat, E., Alshdaifat, D., Alsarhan, A., Hussein, F., & El-Salhi, S. M. F. S. (2021). The effect of preprocessing techniques, applied to numeric features,

on classification algorithms' performance. *Data*, *6*(2), 1–23. https://doi.org/10.3390/data6020011

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00444-8

Ananiadou, S., Pyysalo, S., Tsujii, J., & Kell, D. B. (2010). Event extraction for systems biology by text mining the literature. In *Trends in Biotechnology* (Vol. 28, Issue 7, pp. 381–390). https://doi.org/10.1016/j.tibtech.2010.04.005

Aripin, Agastya, W., & Haryanto, H. (2021). Ekstraksi Emosi Majemuk Kalimat Bahasa Indonesia Menggunakan Convolutional Neural Network. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *10*(2), 148–155. https://doi.org/10.22146/jnteti.v10i2.1051

Arora, S., Liang, Y., & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*.

Asudani, D. S., Nagwani, N. K., & Singh, P. (2022). Exploring the effectiveness of word embedding based deep learning model for improving email classification. *Data Technologies and Applications*, *56*(4), 483–505. https://doi.org/10.1108/DTA-07-2021-0191

Atanbori, J., & Rose, S. (2022). MergedNET: A simple approach for one-shot learning in siamese networks based on similarity layers. *Neurocomputing*, *509*, 1–10. https://doi.org/10.1016/j.neucom.2022.08.070

Augenstein, I., Rocktäschel, T., Vlachos, A., & Bontcheva, K. (2016). Stance detection with bidirectional conditional encoding. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 876–885. https://doi.org/10.18653/v1/d16-1084

Bai, X., Shi, B., Zhang, C., Cai, X., & Qi, L. (2017). Text/non-text image classification in the wild with convolutional neural networks. *Pattern Recognition*, *66*, 437–446. https://doi.org/10.1016/j.patcog.2016.12.005

Baldi, P., & Chauvin, Y. (1993). Neural Networks for Fingerprint Recognition. *Neural Computation*, *5*(3), 402–418. https://doi.org/10.1162/neco.1993.5.3.402

Béchara, H., Costa, H., Taslimipoor, S., Gupta, R., Orăsan, C., Pastor, G. C., & Mitkov, R. (2015). MiniExperts: An SVM Approach for Measuring Semantic Textual Similarity. *SemEval 2015 - 9th International Workshop on Semantic Evaluation, Co-Located with the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2015 - Proceedings, 96–101. https://doi.org/10.18653/v1/s15-2017

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–27. https://doi.org/10.1561/2200000006

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, *3*(6), 1137–1155. https://doi.org/10.1162/153244303322533223

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Beniwal, R., Bhardwaj, D., Raghav, B. P., & Negi, D. (2022). *Text Similarity Identification Based on CNN and CNN-LSTM Model* (pp. 47–58). https://doi.org/10.1007/978-981-16-4641-6_5

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Berlyand, L., & Jabin, P. E. (2023). Mathematics of deep learning: An introduction. In *Mathematics of Deep Learning: An Introduction*. https://doi.org/10.1515/9783111025551

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., & Salakoski, T. (2010). Complex event extraction at PubMed scale. *Bioinformatics*, *26*(12). https://doi.org/10.1093/bioinformatics/btq180

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bouzidi, M. K., & Hashemi, E. (2023). Interaction-Aware Merging in Mixed Traffic with Integrated Game-theoretic Predictive Control and Inverse Differential Game. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2023-June*. https://doi.org/10.1109/IV55152.2023.10186384

Bramer, M. (2007). Principles of Data Mining. In *Principles of Data Mining*. https://doi.org/10.1007/978-1-84628-766-4

Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Säckinger, E., & Shah, R. (1994). *Signature Verification Using a "Siamese" Time Delay Neural Network* (pp. 25–44). https://doi.org/10.1142/9789812797926_0003

Brunila, M., & LaViolette, J. (2021). WMDecompose: A Framework for Leveraging the Interpretable Properties of Word Mover's Distance in Sociocultural Analysis. *5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, LaTeCHCLfL 2021 - Co-Located with the 2021 Conference on Empirical*

Methods in Natural Language Processing, EMNLP 2021 - Proceedings, 154–167. https://doi.org/10.18653/v1/2021.latechclfl-1.18

Buscaldi, D., García Flores, J. J., Meza, I. V., & Rodríguez, I. (2015). SOPA: Random Forests Regression for the Semantic Textual Similarity task. *SemEval 2015 - 9th International Workshop on Semantic Evaluation, Co-Located with the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2015 - Proceedings*, 132–137. https://doi.org/10.18653/v1/s15-2024

Cao, Y., Huang, L., Ji, H., Chen, X., & Li, J. (2017). Bridging text and knowledge by learning multi-prototype entity mention embedding. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 1623–1633. https://doi.org/10.18653/v1/P17-1149

Chen, C. C., Liu, Z., Yang, G., Wu, C. C., & Ye, Q. (2021). An improved fault diagnosis using 1d-convolutional neural network model. *Electronics (Switzerland)*, *10*(1), 1–19. https://doi.org/10.3390/electronics10010059

Chen, L., Chun, L., Ziyu, L., & Quan, Z. (2013). Hybrid pseudo-relevance feedback for microblog retrieval. *Journal of Information Science*, *39*(6), 773–788. https://doi.org/10.1177/0165551513487846

Chen, Q., Du, J., Kim, S., Wilbur, W. J., & Lu, Z. (2020). Deep learning with sentence embeddings pre-trained on biomedical corpora improves the performance of finding similar sentences in electronic medical records. *BMC Medical Informatics and Decision Making*, *20*. https://doi.org/10.1186/s12911-020-1044-0

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 167–176. https://doi.org/10.3115/v1/p15-1017

Chicco, D. (2021). Siamese Neural Networks: An Overview. In *Methods in Molecular Biology* (Vol. 2190, pp. 73–94). https://doi.org/10.1007/978-1-0716-0826-5_3

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. *Proceedings - 2005*

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, *I*, 539–546. https://doi.org/10.1109/CVPR.2005.202

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. http://wordnet.princeton.edu

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Crone, S. F., Lessmann, S., & Stahlbock, R. (2006). The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. *European Journal of Operational Research*, *173*(3), 781–800. https://doi.org/10.1016/j.ejor.2005.07.023

de Souza, J. V. A., e Oliveira, L. E. S., Gumiel, Y. B., Carvalho, D. R., & Moro, C. M. C. (2020). Exploiting siamese neural networks on short text similarity tasks for multiple domains and languages. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12037 LNAI*, 357–367. https://doi.org/10.1007/978-3-030-41505-1_34

Deriu, J., Lucchi, A., De Luca, V., Severyn, A., Muller, S., Cieliebak, M., Hofmann, T., & Jaggi, M. (2017). Leveraging large amounts of weakly supervised data for multi-language sentiment classification. *26th International World Wide Web Conference, WWW 2017*, 1045–1052. https://doi.org/10.1145/3038912.3052611

Di, P., & Duan, L. (2014). New naive Bayes text classification algorithm. *Shuju Caiji Yu Chuli/Journal of Data Acquisition and Processing*, *29*(1), 71–75. https://doi.org/10.11591/telkomnika.v12i2.4180

Dong, L., Wei, F., Zhou, M., & Xu, K. (2015). Question answering over freebase with multi-column convolutional neural networks. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 260–269. https://doi.org/10.3115/v1/p15-1026

Du, M., Chen, W., Liu, K., Wang, L., Hu, Y., Mao, Y., Sun, X., Luo, Y., Shi, J., Shao, K., Huang, H., & Ye, D. (2022). The Global Burden of Leukemia and Its Attributable Factors in 204 Countries and Territories: Findings from the Global Burden of Disease 2019 Study and Projections to 2030. *Journal of Oncology*, *2022*. https://doi.org/10.1155/2022/1612702

Fan, H., Jiang, M., Xu, L., Zhu, H., Cheng, J., & Jiang, J. (2020). Comparison of

long short term memory networks and the hydrological model in runoff simulation. *Water (Switzerland)*, *12*(1). https://doi.org/10.3390/w12010175

Farouk, M. (2018). Sentence Semantic Similarity based on Word Embedding and WordNet. *Proceedings - 2018 13th International Conference on Computer Engineering and Systems, ICCES 2018*, 33–37. https://doi.org/10.1109/ICCES.2018.8639211

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 1606–1615. https://doi.org/10.3115/v1/n15-1184

Feldmen, R., & Sanger, J. (2007). The text mining handbook: advanced approaches in analyzing unstructured data. *Choice Reviews Online*, *44*(10), 44-5684-44–5684. https://doi.org/10.5860/choice.44-5684

Fernández, J., Gutiérrez, Y., Gómez, J. M., & Martínez-Barco, P. (2014). GPLSI: Supervised Sentiment Analysis in Twitter using Skipgrams. *8th International Workshop on Semantic Evaluation, SemEval 2014 - Co-Located with the 25th International Conference on Computational Linguistics, COLING 2014, Proceedings*, 294–299. https://doi.org/10.3115/v1/s14-2048

Firmansyah, M. R., Ilyas, R., & Kasyidi, F. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network. *Prosiding The 11th Industrial Research Workshop and National Seminar*, *11*(1), 488–495.

Ganguly, D., Roy, D., Mitra, M., & Jones, G. J. F. (2015). A word embedding based generalized language model for information retrieval. *SIGIR 2015 - Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 795–798. https://doi.org/10.1145/2766462.2767780

Gao, M., Li, T., & Huang, P. (2019). Text classification research based on improved word2vec and CNN. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11434 LNCS*, 126–135. https://doi.org/10.1007/978-3-030-17642-6_11

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation*, *12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, *57*, 345–420. https://doi.org/10.1613/jair.4992

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. In *Nature* (Vol. 29, Issue 7553, pp. 1–73).

Goudjil, M., Koudil, M., Bedda, M., & Ghoggali, N. (2018). A Novel Active Learning Method Using SVM for Text Classification. *International Journal of Automation and Computing*, *15*(3), 290–298. https://doi.org/10.1007/s11633-015-0912-z

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855–868. https://doi.org/10.1109/TPAMI.2008.137

Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

Gu, W., Yang, X., Yang, M., Han, K., Pan, W., & Zhu, Z. (2022). MarkerGenie: an NLP-enabled text-mining system for biomedical entity relation extraction. *Bioinformatics Advances*, *2*(1). https://doi.org/10.1093/bioadv/vbac035

Gudivada, V. N., & Arbabifard, K. (2018). Open-Source Libraries, Application Frameworks, and Workflow Systems for NLP. In *Handbook of Statistics* (Vol. 38, pp. 31–50). https://doi.org/10.1016/bs.host.2018.07.007

Gurulingappa, H., Bauer, A., Toldo, L., Schepers, C., & Megaro, G. (2016). Semi-Supervised Information Retrieval System for Clinical Decision Support. *25th Text REtrieval Conference, TREC 2016 - Proceedings*, 1–10. http://trec.nist.gov/pubs/trec25/papers/MERCKKGAA-CL.pdf

Gurusamy, R., & Subramaniam, V. (2017). A machine learning approach for MRI brain tumor classification. *Computers, Materials and Continua*, *53*(2), 91–109.

Ha, J., Kambe, M., & Pe, J. (2011). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. https://doi.org/10.1016/C2009-0-61819-5

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. In *Expert Systems with Applications* (Vol. 73, pp. 220–239). https://doi.org/10.1016/j.eswa.2016.12.035

Han, J., Kamber, M., & Pei, J. (2012). Getting to Know Your Data. In *Data Mining* (pp. 39–82). https://doi.org/10.1016/b978-0-12-381479-1.00002-2

Han, X., Liu, Z., & Sun, M. (2016). *Joint Representation Learning of Text and Knowledge for Knowledge Graph Completion*. http://arxiv.org/abs/1611.04125

Hasib, K. M., Azam, S., Karim, A., Marouf, A. Al, Shamrat, F. M. J. M., Montaha, S., Yeo, K. C., Jonkman, M., Alhajj, R., & Rokne, J. G. (2023). MCNN-

LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data. *IEEE Access*, *11*, 93048–93063. https://doi.org/10.1109/ACCESS.2023.3309697

Hassan, H., Ahmad, N. B., & Anuar, S. (2020). Improved students' performance prediction for multi-class imbalanced problems using hybrid and ensemble approach in educational data mining. *Journal of Physics: Conference Series*, *1529*(5). https://doi.org/10.1088/1742-6596/1529/5/052041

Hayashi, T., & Fujita, H. (2022). One-class ensemble classifier for data imbalance problems. *Applied Intelligence*, *52*(15), 17073–17089. https://doi.org/10.1007/s10489-021-02671-1

He, H., & Lin, J. (2016). Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 937–948. https://doi.org/10.18653/v1/n16-1108

He, H., & Ma, Y. (2013). Imbalanced learning: Foundations, algorithms, and applications. In *Imbalanced Learning: Foundations, Algorithms, and Applications*. https://doi.org/10.1002/9781118646106

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, *98*, 105–117. https://doi.org/10.1016/j.eswa.2018.01.012

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data*, *9*(1). https://doi.org/10.1186/s40537-022-00652-w

Ilina, O., Ziyadinov, V., Klenov, N., & Tereshonok, M. (2022). A Survey on Symmetrical Neural Network Architectures and Applications. In *Symmetry* (Vol. 14, Issue 7). https://doi.org/10.3390/sym14071391

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A Review of Deep Transfer Learning and Recent Advancements. In *Technologies* (Vol. 11, Issue 2). https://doi.org/10.3390/technologies11020040

Iyyer, M., Manjunatha, V., Boyd-Graber, J., & Daumé, H. (2015). Deep unordered composition rivals syntactic methods for text classification. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 1681–1691. https://doi.org/10.3115/v1/p15-1162

Jagannatha, A. N., Chen, J., & Yu, H. (2015). Mining and Ranking Biomedical Synonym Candidates from Wikipedia. *EMNLP 2015 - 6th International Workshop on Health Text Mining and Information Analysis, LOUHI 2015 - Proceedings of the Workshop*, 142–151. https://doi.org/10.18653/v1/w15-2619

Jana, M., & Biswas, S. (2021). Intelligent and smart enabling technologies in advanced applications: Recent trends. In *Recent Trends in Computational Intelligence Enabled Research: Theoretical Foundations and Applications* (pp. 355–365). https://doi.org/10.1016/B978-0-12-822844-9.00045-1

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/s12525-021-00475-2

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449. https://doi.org/10.3233/ida-2002-6504

Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in English words. *Procedia Computer Science*, *157*, 160–167. https://doi.org/10.1016/j.procs.2019.08.153

Jha, A., Dave, M., & Madan, S. (2019). Comparison of Binary Class and Multi-Class Classifier Using Different Data Mining Classification Techniques. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3464211

Jian, C., Gao, J., & Ao, Y. (2016). A new sampling method for classifying imbalanced data based on support vector machine ensemble. *Neurocomputing*, *193*, 115–122. https://doi.org/10.1016/j.neucom.2016.02.006

Jiang, P., & Chen, J. (2016). Displacement prediction of landslide based on generalized regression neural networks with K-fold cross-validation. *Neurocomputing*, *198*, 40–47. https://doi.org/10.1016/j.neucom.2015.08.118

Jiang, Z., Jin, L. K., Li, L. S., Qin, M., Qu, C., Zheng, J., & Huang, D. (2015). A CRD-WEL system for chemical-disease relations extraction. *The Fifth BioCreative Challenge Evaluation Workshop*, 317–326.

Jiang, Z., Li, L., & Huang, D. (2016). A general protein-protein interaction extraction architecture based on word representation and feature selection. *International Journal of Data Mining and Bioinformatics*, *14*(3), 276–291. https://doi.org/10.1504/IJDMB.2016.074878

Jiao, J. (2020). Application and prospect of artificial intelligence in smart grid. *IOP Conference Series: Earth and Environmental Science*, *510*(2). https://doi.org/10.1088/1755-1315/510/2/022012

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Deep learning and thresholding with class-imbalanced big data. *Proceedings - 18th IEEE International Conference*

on Machine Learning and Applications, ICMLA 2019, 755–762. https://doi.org/10.1109/ICMLA.2019.00134

Joshi, A. V. (2020). Introduction to AI and ML. In *Machine Learning and Artificial Intelligence* (pp. 3–7). https://doi.org/10.1007/978-3-030-26622-6_1

Jp, S., Menon, V. K., Kp, S., S, R., & Wolk, A. (2021). Generation of cross-lingual word vectors for low-resourced languages using deep learning and topological metrics in a data-efficient way. *Electronics (Switzerland)*, *10*(12). https://doi.org/10.3390/electronics10121372

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, *1*, 655–665. https://doi.org/10.3115/v1/p14-1062

Karhunen, J., Raiko, T., & Cho, K. H. (2015). Unsupervised deep learning: A short review. In *Advances in Independent Component Analysis and Learning Machines* (pp. 125–142). https://doi.org/10.1016/B978-0-12-802806-3.00007-5

Kashyap, A., Han, L., Yus, R., Sleeman, J., Satyapanich, T., Gandhi, S., & Finin, T. (2016). Robust semantic text similarity using LSA, machine learning, and linguistic resources. *Language Resources and Evaluation*, *50*(1), 125–161. https://doi.org/10.1007/s10579-015-9319-2

Kenter, T., & De Rijke, M. (2015). Short text similarity with word embeddings. *International Conference on Information and Knowledge Management, Proceedings*, *19-23-Oct-*, 1411–1420. https://doi.org/10.1145/2806416.2806475

Khan, R., Akbar, S., Mehmood, A., Shahid, F., Munir, K., Ilyas, N., Asif, M., & Zheng, Z. (2023). A transfer learning approach for multiclass classification of Alzheimer's disease using MRI images. *Frontiers in Neuroscience*, *16*. https://doi.org/10.3389/fnins.2022.1050777

Kilimci, Z. H., & Akyokus, S. (2018). Deep learning- and word embedding-based heterogeneous classifier ensembles for text classification. *Complexity*, *2018*. https://doi.org/10.1155/2018/7130146

Kim, I. J., & Xie, X. (2015). Handwritten Hangul recognition using deep convolutional neural networks. *International Journal on Document Analysis and Recognition*, *18*(1), 1–13. https://doi.org/10.1007/s10032-014-0229-4

Kim, J. D., Ohta, T., & Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, *9*. https://doi.org/10.1186/1471-2105-9-10

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language*

Processing, Proceedings of the Conference, 1746–1751. https://doi.org/10.3115/v1/d14-1181

Korde, V. (2012). Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85–99. https://doi.org/10.5121/ijaia.2012.3208

Kotei, E., & Thirunavukarasu, R. (2023). A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. In *Information (Switzerland)* (Vol. 14, Issue 3). https://doi.org/10.3390/info14030187

Kowsari, K., Brown, D. E., Heidarysafa, M., Jafari Meimandi, K., Gerber, M. S., & Barnes, L. E. (2017). HDLTex: Hierarchical Deep Learning for Text Classification. *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, *2017-Decem*, 364–371. https://doi.org/10.1109/ICMLA.2017.0-134

Krishnan, K., & Rogers, S. P. (2015). A New Universe of Data. In *Social Data Analytics* (pp. 1–10). https://doi.org/10.1016/b978-0-12-397186-9.00001-7

Kumar, P., Joy, J., Pandey, A., & Gupta, D. (2017). PRmePRed: A protein arginine methylation prediction tool. *PLoS ONE*, *12*(8). https://doi.org/10.1371/journal.pone.0183318

L'Heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine Learning with Big Data: Challenges and Approaches. *IEEE Access*, *5*, 7776–7797. https://doi.org/10.1109/ACCESS.2017.2696365

Ladani, D. J., & Desai, N. P. (2020). Stopword Identification and Removal Techniques on TC and IR applications: A Survey. *2020 6th International Conference on Advanced Computing and Communication Systems, ICACCS 2020*, 466–472. https://doi.org/10.1109/ICACCS48705.2020.9074166

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. *Proceedings of 2016 4th International Conference on Cyber and IT Service Management, CITSM 2016*. https://doi.org/10.1109/CITSM.2016.7577578

Le, X. H., Viet, H. H., Lee, G., Le, X.-H., & Ho, H. V. (2019). River streamflow prediction using a deep neural network: a case study on the Red River, Vietnam. *Agricultural Science Korean Journal of Agricultural Science*, *46*(4). https://www.researchgate.net/publication/340095460

LeCun, Y., Hinton, G., & Bengio, Y. (2015). Deep learning (2015), Y. LeCun, Y. Bengio and G. Hinton. *Nature*, *521*, 436–444.

Lee, R. S. T. (2020). Artificial Intelligence in Daily Life. In *Artificial Intelligence in Daily Life*. https://doi.org/10.1007/978-981-15-7695-9

Levy, O., & Goldberg, Y. (2014). Dependency-based word embeddings. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, *2*, 302–308. https://doi.org/10.3115/v1/p14-2050

Levy, O., Kenett, Y. N., Oxenberg, O., Castro, N., De Deyne, S., Vitevitch, M. S., & Havlin, S. (2021). Unveiling the nature of interaction between semantics and phonology in lexical access based on multilayer networks. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-93925-y

Li, H., & Xu, J. (2013). Semantic matching in search. *Foundations and Trends in Information Retrieval*, *7*(5), 343–469. https://doi.org/10.1561/1500000035

Li, M., Zhou, X., Ryu, K. H., & Theera-Umpon, N. (2022). An Ensemble Semantic Textual Similarity Measure Based on Multiple Evidences for Biomedical Documents. *Computational and Mathematical Methods in Medicine*, *2022*. https://doi.org/10.1155/2022/8238432

Li, Q., & He, S. (2023). Similarity matching of medical question based on Siamese network. *BMC Medical Informatics and Decision Making*, *23*(1). https://doi.org/10.1186/s12911-023-02161-z

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P. S., & He, L. (2022). A Survey on Text Classification: From Traditional to Deep Learning. In *ACM Transactions on Intelligent Systems and Technology* (Vol. 13, Issue 2). https://doi.org/10.1145/3495162

Li, S., & Gong, B. (2021). Word embedding and text classification based on deep learning methods. *MATEC Web of Conferences*, *336*, 06022. https://doi.org/10.1051/matecconf/202133606022

Li, Y., & Yang, T. (2018). Word Embedding for Understanding Natural Language: A Survey. In *Studies in Big Data* (Vol. 26, pp. 83–104). https://doi.org/10.1007/978-3-319-53817-4_4

Li, Y., Zhang, Y., & Cai, Y. (2021). A new hyper-parameter optimization method for power load forecast based on recurrent neural networks. *Algorithms*, *14*(6). https://doi.org/10.3390/a14060163

Lin, C., Huang, Z., Yang, F., & Zou, Q. (2012). Identify content quality in online social networks. *IET Communications*, *6*(12), 1618–1624. https://doi.org/10.1049/iet-com.2011.0202

Liu, H., Zhou, M., & Liu, Q. (2019). An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, *6*(3), 703–715. https://doi.org/10.1109/JAS.2019.1911447

Liu, S., Tang, B., Chen, Q., & Wang, X. (2016). Drug-Drug Interaction Extraction via Convolutional Neural Networks. *Computational and Mathematical Methods in Medicine*, *2016*. https://doi.org/10.1155/2016/6918381

Liu, T., Bao, J., Wang, J., & Zhang, Y. (2018). A hybrid CNN-LSTM algorithm for online defect recognition of CO2 welding. *Sensors (Switzerland)*, *18*(12). https://doi.org/10.3390/S18124369

Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Medical Research Methodology*, *22*(1). https://doi.org/10.1186/s12874-022-01665-y

Lumbanraja, F. R., Mahesworo, B., Cenggoro, T. W., Sudigyo, D., & Pardamean, B. (2021). SSMFN: a fused spatial and sequential deep learning model for methylation site prediction. *PeerJ Computer Science*, *7*, 1–14. https://doi.org/10.7717/peerj-cs.683

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *1*, 142–150.

Malik, S., Shoaib, U., Bukhari, S. A. C., El Sayed, H., & Khan, M. A. (2022). A hybrid query expansion framework for the optimal retrieval of the biomedical literature. *Smart Health*, *23*. https://doi.org/10.1016/j.smhl.2021.100247

Mangat, P. K., & Saini, K. S. (2022). Relevance of data mining techniques in real life. In *System Assurances: Modeling and Management* (pp. 477–502). https://doi.org/10.1016/B978-0-323-90240-3.00026-6

Mansoor, M., Ur Rehman, Z., Shaheen, M., Khan, M. A., & Habib, M. (2020). Deep learning based semantic similarity detection using text data. *Information Technology and Control*, *49*(4), 495–510. https://doi.org/10.5755/j01.itc.49.4.27118

Markoulidakis, I., Rallis, I., Georgoulas, I., Kopsiaftis, G., Doulamis, A., & Doulamis, N. (2021). Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem. *Technologies*, *9*(4). https://doi.org/10.3390/technologies9040081

Mehndiratta, P., & Soni, D. (2019). Identification of Sarcasm in Textual Data: A Comparative Study. *Journal of Data and Information Science*, *4*(4), 56–83. https://doi.org/10.2478/jdis-2019-0021

Metzler, D., Dumais, S., & Meek, C. (2007). Similarity measures for short segments of text. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4425 LNCS*, 16–27. https://doi.org/10.1007/978-3-540-71496-5_5

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In: Conference on Advances in Neural Information Processing Systems. *Distributed Representations of Words and Phrases and Their Compositionality*.

Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. *Proceedings of the 2nd Workshop on Computational Linguistics for Literature, CLfL 2013 at the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2013*, 746–751.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning-Based Text Classification. In *ACM Computing Surveys* (Vol. 54, Issue 3). https://doi.org/10.1145/3439726

Miner, G., Delen, D., Elder, J., Fast, A., Hill, T., & Nisbet, R. A. (2012). Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. In *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. https://doi.org/10.1016/C2010-0-66188-8

Mnih, A., & Kavukcuoglu, K. (2013). Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*.

Mohamed, E. H., Moussa, M. E. S., & Haggag, M. H. (2020). An Enhanced Sentiment Analysis Framework Based on Pre-Trained Word Embedding. *International Journal of Computational Intelligence and Applications*, *19*(4). https://doi.org/10.1142/S1469026820500315

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, 243–248. https://doi.org/10.1109/ICICS49469.2020.239556

Morimoto, J., Ponchon, A., Sofronov, G., & Travis, J. (2021). Editorial: Applications of Machine Learning to Evolutionary Ecology Data. In *Frontiers in Ecology and Evolution* (Vol. 9). https://doi.org/10.3389/fevo.2021.797319

Mueller, J., & Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2786–2792. https://doi.org/10.1609/aaai.v30i1.10350

Musolf, A. M., Holzinger, E. R., Malley, J. D., & Bailey-Wilson, J. E. (2022). What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics. In *Human Genetics* (Vol. 141, Issue 9, pp. 1515–1528). https://doi.org/10.1007/s00439-021-02402-z

Neto, J., Santos, A., & Kaestner, C. (2000). Document clustering and text summarization. *Proceedings of the 4th International Conference {...}*. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4634%5Cnhttp://kar.kent.ac.uk/21916/

Nguyen, H. T., Duong, P. H., & Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowledge-Based Systems*, *182*. https://doi.org/10.1016/j.knosys.2019.07.013

Nguyen, T. H., & Grishman, R. (2014). Employing word representations and regularization for domain adaptation of relation extraction. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, *2*, 68–74. https://doi.org/10.3115/v1/p14-2012

Nikhath, A. K., Subrahmanyam, K., & Vasavi, R. (2016). Building a K-Nearest Neighbor Classifier for Text Categorization. *(IJCSIT) International Journal of Computer Science and Information Technologies*, *7*(1), 254–256. www.ijcsit.com

Niraula, N., Banjade, R., Ştefănescu, D., & Rus, V. (2013). Experiments with semantic similarity measures based on LDA and LSA. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7978 LNAI*, 188–199. https://doi.org/10.1007/978-3-642-39593-2_17

Notonogoro, I. W., Jondri, & Arifianto, A. (2018). Indonesian license plate recognition using convolutional neural network. *2018 6th International Conference on Information and Communication Technology, ICoICT 2018*, 366–369. https://doi.org/10.1109/ICoICT.2018.8528761

Nurdin, A., & Maulidevi, N. U. (2018). 5W1H Information Extraction with CNN-Bidirectional LSTM. *Journal of Physics: Conference Series*, *978*(1). https://doi.org/10.1088/1742-6596/978/1/012078

Ochieng, P. J., & Djatna, T. (2014). *Media Monitoring Analysis Based on Text Mining Clustering Using Pillar K-means Algorithm*. 1–6. https://doi.org/10.13140/RG.2.1.4342.1526

Olivas, E. S., Guerrero, J. D. M., Martinez Sober, M., Magdalena Benedito, J. R., & Serrano López, A. J. (2009). Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. https://doi.org/10.4018/978-1-60566-766-9

Oluwaseun, A., & Chaubey, M. S. (2019). Data Mining Classification Techniques on the analysis of student performance. *Global Scientific Journal*, *7*(April), 79–95. https://doi.org/10.11216/gsj.2019.04.19671

Othman, N., Faiz, R., & Smaïli, K. (2019). Manhattan siamese LSTM for question

retrieval in community question answering. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11877 LNCS*, 661–677. https://doi.org/10.1007/978-3-030-33246-4_41

Panesar, A. (2019). What Is Machine Learning? In *Machine Learning and AI for Healthcare* (pp. 75–118). Springer International Publishing. https://doi.org/10.1007/978-1-4842-3799-1_3

Paraskevopoulos, S., Smeets, P., Tian, X., & Medema, G. (2022). Using Artificial Intelligence to extract information on pathogen characteristics from scientific publications. *International Journal of Hygiene and Environmental Health*, *245*. https://doi.org/10.1016/j.ijheh.2022.114018

Parwez, M. A., Fazil, M., Arif, M., Nafis, M. T., & Auwul, M. R. (2023). Biomedical Text Classification Using Augmented Word Representation Based on Distributional and Relational Contexts. *Computational Intelligence and Neuroscience*, *2023*, 1–22. https://doi.org/10.1155/2023/2989791

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Pontes, E. L., Huet, S., Linhares, A. C., & Torres-Moreno, J.-M. (2018). *Predicting the Semantic Textual Similarity with Siamese CNN and LSTM*. TALN. http://arxiv.org/abs/1810.10641

Prabhakar, S. K., Rajaguru, H., & Won, D. O. (2021). Performance Analysis of Hybrid Deep Learning Models with Attention Mechanism Positioning and Focal Loss for Text Classification. *Scientific Programming*, *2021*. https://doi.org/10.1155/2021/2420254

Prakoso, D. W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: a review. *Soft Computing*, *25*(6), 4699–4723. https://doi.org/10.1007/s00500-020-05479-2

Prasetya, D. D., Wibawa, A. P., & Hirashima, T. (2018). The performance of text similarity algorithms. *International Journal of Advances in Intelligent Informatics*, *4*(1), 63–69. https://doi.org/10.26555/ijain.v4i1.152

Pribadi, N. H., Sarno, R., Ahmadiyah, A. S., & Sungkono, K. R. (2021). Semantic Recommender System Based on Semantic Similarity Using FastText and Word Mover's Distance. *International Journal of Intelligent Engineering and Systems*, *14*(2), 377–385. https://doi.org/10.22266/ijies2021.0430.34

Pristyanto, Y., Pratama, I., & Nugraha, A. F. (2018). Data level approach for imbalanced class handling on educational data mining multiclass classification. *2018 International Conference on Information and*

Communications Technology, ICOIACT 2018, *2018-Janua*, 310–314. https://doi.org/10.1109/ICOIACT.2018.8350792

Qamar, U., & Raza, M. S. (2020). Data Science Concepts and Techniques with Applications. In *Data Science Concepts and Techniques with Applications*. https://doi.org/10.1007/978-981-15-6133-7

Rabut, B. A., Fajardo, A. C., & Medina, R. P. (2019). Multi-class document classification using improved word embeddings. *ACM International Conference Proceeding Series*, 42–46. https://doi.org/10.1145/3366650.3366661

Rafaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-Validation in Encyclopedia of Database System. In *SAE International* (pp. 532–538). path://ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/%5Cn0000875816.3D

Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012). Semantic Cosine Similarity. *The 7th International Student Conference on Advanced Science and Technology ICAST*, *4*(1). https://www.researchgate.net/publication/262525676

Rajesh, K. N. V. P. S., & Dhuli, R. (2018). Classification of imbalanced ECG beats using re-sampling techniques and AdaBoost ensemble classifier. *Biomedical Signal Processing and Control*, *41*, 242–254. https://doi.org/10.1016/j.bspc.2017.12.004

Ranasinghe, T., Orasan, C., & Mitkov, R. (2019). Semantic textual similarity with Siamese neural networks. *International Conference Recent Advances in Natural Language Processing, RANLP*, *2019-Septe*, 1004–1011. https://doi.org/10.26615/978-954-452-056-4_116

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. In *Neural Computation* (Vol. 29, Issue 9, pp. 2352–2449). https://doi.org/10.1162/NECO_a_00990

Redd, M. V., & M. Hanumanthappa. (2014). Semantical and Syntactical Analysis of NLP. *(IJCSIT) International Journal of Computer Science and Information Technologies*, *5*(3), 3236–3238. http://www.universalteacher.org.uk/lang/semantics.html

Rehman, A. U., Malik, A. K., Raza, B., & Ali, W. (2019). A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. *Multimedia Tools and Applications*, *78*(18), 26597–26613. https://doi.org/10.1007/s11042-019-07788-7

Reimers, N., Beyer, P., & Gurevych, I. (2016). Task-oriented intrinsic evaluation of semantic textual similarity. *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical*

Papers, 87–96.

Ren, F., Cao, P., Li, W., Zhao, D., & Zaiane, O. (2017). Ensemble based adaptive over-sampling method for imbalanced data learning in computer aided detection of microaneurysm. *Computerized Medical Imaging and Graphics*, *55*, 54–67. https://doi.org/10.1016/j.compmedimag.2016.07.011

Ren, M., Kiros, R., & Zemel, R. S. (2015). Exploring models and data for image question answering. *Advances in Neural Information Processing Systems*, *2015-Janua*, 2953–2961.

Rifat, R. H., Poran, M. S., Islam, S., Sumaya, A. T., Alam, M. M., & Rahman, M. R. (2023). Incidence, Mortality, and Epidemiology of Leukemia in South Asia: An Ecological Study. *Open Journal of Epidemiology*, *13*(01), 73–82. https://doi.org/10.4236/ojepi.2023.131006

Rios, A., & Lwowski, B. (2020). An Empirical Study of the Downstream Reliability of Pre-Trained Word Embeddings. *COLING 2020 - 28th International Conference on Computational Linguistics, Proceedings of the Conference*, 3371–3388. https://doi.org/10.18653/v1/2020.coling-main.299

Rohani, A., Taki, M., & Abdollahpour, M. (2018). A novel soft computing model (Gaussian process regression with K-fold cross validation) for daily and monthly solar radiation forecasting (Part: I). *Renewable Energy*, *115*, 411–422. https://doi.org/10.1016/j.renene.2017.08.061

Romanov, A., & Shivade, C. (2018). Lessons from natural language inference in the clinical domain. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 1586–1596. https://doi.org/10.18653/v1/d18-1187

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for sentence summarization. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, *1509*(685), 379–389.

Saeed, A., Shoukat, S., Shehzad, K., Ahmad, I., Eshmawi, A. A., Amin, A. H., & Tag-Eldin, E. (2022). A Deep Learning-Based Approach for the Diagnosis of Acute Lymphoblastic Leukemia. *Electronics (Switzerland)*, *11*(19). https://doi.org/10.3390/electronics11193168

Salman, U. T., Rehman, S., Alawode, B., & Alhems, L. M. (2021). Short Term Prediction of Wind Speed Based on Long-Short Term Memory Networks. *FME Transactions*, *49*(3), 643–652. https://doi.org/10.5937/fme2103643S

Sarker, I. H. (2021a). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 6). https://doi.org/10.1007/s42979-021-00815-1

Sarker, I. H. (2021b). Machine Learning: Algorithms, Real-World Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 3).

https://doi.org/10.1007/s42979-021-00592-x

Serrano, N., & Bellogín, A. (2023). Siamese neural networks in recommendation. In *Neural Computing and Applications* (Vol. 35, Issue 19, pp. 13941–13953). https://doi.org/10.1007/s00521-023-08610-0

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research*, *5*(1). https://doi.org/10.1007/s41133-020-00032-0

Shahi, T. B., Sitaula, C., & Paudel, N. (2022). A Hybrid Feature Extraction Method for Nepali COVID-19-Related Tweets Classification. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/5681574

Shaikh, S., Daudpota, S. M., Imran, A. S., & Kastrati, Z. (2021). Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models. *Applied Sciences (Switzerland)*, *11*(2), 1–20. https://doi.org/10.3390/app11020869

Shajalal, M., & Aono, M. (2019). Semantic textual similarity between sentences using bilingual word semantics. *Progress in Artificial Intelligence*, *8*(2), 263–272. https://doi.org/10.1007/s13748-019-00180-4

Shajalal, M., Ullah, M. Z., Chy, A. N., & Aono, M. (2016). Query subtopic diversification based on cluster ranking and semantic features. *4th IGNITE Conference and 2016 International Conference on Advanced Informatics: Concepts, Theory and Application, ICAICTA 2016*. https://doi.org/10.1109/ICAICTA.2016.7803099

Shao, Y. (2017). Use convolutional neural network to evaluate Semantic Textual Similarity. *Proceedings Ofthe 11th International Workshop on Semantic Evaluations (SemEval-2017)*, 130–133. http://ixa2.si.ehu.es/stswiki/index.php/STSbenchmark

Sharma, R., & Jani, C. (2022). Mapping incidence and mortality of leukemia and its subtypes in 21 world regions in last three decades and projections to 2030. *Annals of Hematology*, *101*(7), 1523–1534. https://doi.org/10.1007/s00277-022-04843-6

Sharma, R., Palshikar, G., & Pawar, S. (2018). An unsupervised approach for cause-effect relation extraction from biomedical text. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10859 LNCS*, 419–427. https://doi.org/10.1007/978-3-319-91947-8_43

Shen, F., & Lee, Y. (2016). Knowledge discovery from biomedical ontologies in cross domains. *PLoS ONE*, *11*(8). https://doi.org/10.1371/journal.pone.0160005

Shen, F., Liu, H., Sohn, S., Larson, D. W., & Lee, Y. (2015). BmQGen: Biomedical query generator for knowledge discovery. *Proceedings - 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015*, 1092–1097. https://doi.org/10.1109/BIBM.2015.7359833

Shen, F., Liu, H., Sohn, S., Larson, D. W., & Lee, Y. (2016). Predicate Oriented Pattern Analysis for Biomedical Knowledge Discovery. *Intelligent Information Management*, *08*(03), 66–85. https://doi.org/10.4236/iim.2016.83006

Shi, H., Wang, C., & Sakai, T. (2020a). A {S}iamese {CNN} Architecture for Learning {C}hinese Sentence Similarity. *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, 24–29. https://aclanthology.org/2020.aacl-srw.4

Shi, H., Wang, C., & Sakai, T. (2020b). A Siamese CNN Architecture for Learning Chinese Sentence Similarity. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*. https://aclanthology.org/2020.aacl-srw.4

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1631–1642.

Soltanzadeh, P., Feizi-Derakhshi, M. R., & Hashemzadeh, M. (2023). Addressing the class-imbalance and class-overlap problems by a metaheuristic-based under-sampling approach. *Pattern Recognition*, *143*. https://doi.org/10.1016/j.patcog.2023.109721

Sriganesh, V. (2011). Using PubMed in radiology: Ten useful tips for radiologists. *Indian Journal of Radiology and Imaging*, *21*(3), 162–169. https://doi.org/10.4103/0971-3026.85362

Srinivasan, V., Santhanam, S., & Shaikh, S. (2021). Using reinforcement learning with external rewards for open-domain natural language generation. *Journal of Intelligent Information Systems*, *56*(1), 189–206. https://doi.org/10.1007/s10844-020-00626-5

Stanojevic, M., Alshehri, J., & Obradovic, Z. (2021). *High Performance Computing for Understanding Natural Language* (pp. 133–144). https://doi.org/10.4018/978-1-7998-7156-9.ch010

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), 77–89. https://doi.org/10.1016/S0034-4257(97)00083-7

Sunilkumar, P., & Shaji, A. P. (2019). A Survey on Semantic Similarity. *2019 6th IEEE International Conference on Advances in Computing, Communication and Control, ICAC3 2019*. https://doi.org/10.1109/ICAC347590.2019.9036843

Tang, B., Cao, H., Wang, X., Chen, Q., & Xu, H. (2014). Evaluating word representation features in biomedical named entity recognition tasks. *BioMed Research International*, *2014*. https://doi.org/10.1155/2014/240403

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, *1*, 1555–1565. https://doi.org/10.3115/v1/p14-1146

Tien, N. H., Le, N. M., Tomohiro, Y., & Tatsuya, I. (2019). Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity. *Information Processing and Management*, *56*(6). https://doi.org/10.1016/j.ipm.2019.102090

Torregrossa, F., Allesiardo, R., Claveau, V., Kooli, N., & Gravier, G. (2021). A survey on training and evaluation of word embeddings. In *International Journal of Data Science and Analytics* (Vol. 11, Issue 2, pp. 85–103). https://doi.org/10.1007/s41060-021-00242-8

Tran, T. T. T., Nghiem, S. V., Le, V. T., Quan, T. T., Nguyen, V., Yip, H. Y., & Bodenreider, O. (2020). Siamese KG-LSTM: A deep learning model for enriching UMLS Metathesaurus synonymy. *Proceedings - 2020 12th International Conference on Knowledge and Systems Engineering, KSE 2020*, 281–286. https://doi.org/10.1109/KSE50997.2020.9287797

Triguero, I., & Galar, M. (2023). Machine Learning with Spark. In *Large-Scale Data Analytics with Python and Spark*. https://doi.org/10.1017/9781009318242.007

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, *2010-July*, 384–394.

Usman, S., Mehmood, R., Katib, I., & Albeshri, A. (2023). Data Locality in High Performance Computing, Big Data, and Converged Systems: An Analysis of the Cutting Edge and a Future System Architecture. In *Electronics (Switzerland)* (Vol. 12, Issue 1). https://doi.org/10.3390/electronics12010053

Vayena, E., & Blasimme, A. (2017). Biomedical Big Data: New Models of Control Over Access, Use and Governance. *Journal of Bioethical Inquiry*, *14*(4), 501–513. https://doi.org/10.1007/s11673-017-9809-6

Vijayalakshmi, V., & Venkatachalapathy, K. (2019). Deep neural network for multi-class prediction of student performance in educational data. *International Journal of Recent Technology and Engineering*, *8*(2), 5073–5081. https://doi.org/10.35940/ijrte.B2155.078219

Vinet, L., & Zhedanov, A. (2011). A “missing” family of classical orthogonal polynomials. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8). https://doi.org/10.1088/1751-8113/44/8/085201

Wang, C., Nulty, P., & Lillis, D. (2020). A Comparative Study on Word Embeddings in Deep Learning for Text Classification. *ACM International Conference Proceeding Series*, 37–46. https://doi.org/10.1145/3443279.3443304

Wang, H., & Li, F. (2022). A text classification method based on LSTM and graph attention network. *Connection Science*, *34*(1), 2466–2480. https://doi.org/10.1080/09540091.2022.2128047

Wang, J., & Dong, Y. (2020). Measurement of text similarity: A survey. In *Information (Switzerland)* (Vol. 11, Issue 9, pp. 1–17). https://doi.org/10.3390/info11090421

Wang, Y., Afzal, N., Fu, S., Wang, L., Shen, F., Rastegar-Mojarad, M., & Liu, H. (2020). MedSTS: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, *54*(1), 57–72. https://doi.org/10.1007/s10579-018-9431-1

Wang, Y., Fu, S., Shen, F., Henry, S., Uzuner, O., & Liu, H. (2020). The 2019 n2c2/OHNLP track on clinical semantic textual similarity: Overview. *JMIR Medical Informatics*, *8*(11). https://doi.org/10.2196/23375

Wang, Y., Rastegar-Mojarad, M., Komandur-Elayavilli, R., Liu, S., & Liu, H. (2016). An Ensemble Model of Clinical Information Extraction and Information Retrieval for Clinical Decision Support. *25th Text REtrieval Conference, TREC 2016 - Proceedings*.

Wang, Y., Verspoor, K., & Baldwin, T. (2020). *Learning from Unlabelled Data for Clinical Semantic Textual Similarity*. 227–233. https://doi.org/10.18653/v1/2020.clinicalnlp-1.25

Wang, Y., Wang, L., Liu, S., Shen, F., Liu, H., Rastegar-Mojarad, M., & Liu, F. (2017). Dependency and AMR embeddings for drug-drug interaction extraction from biomedical literature. *ACM-BCB 2017 - Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 36–43. https://doi.org/10.1145/3107411.3107426

Weeks, G. R. (1864). Pyæmia, or Leukæmia. *The Boston Medical and Surgical Journal*, *69*(23), 459–463. https://doi.org/10.1056/nejm186401070692305

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2016). Charagram: Embedding words and sentences via character n-grams. *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 1504–1515. https://doi.org/10.18653/v1/d16-1157

Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information (Switzerland)*, *14*(1). https://doi.org/10.3390/info14010054

Wu, Y., Xu, J., Zhang, Y., & Xu, H. (2015). Clinical Abbreviation Disambiguation Using Neural Word Embeddings. *ACL-IJCNLP 2015 - BioNLP 2015: Workshop on Biomedical Natural Language Processing, Proceedings of the Workshop*, 171–176. https://doi.org/10.18653/v1/w15-3822

Xu, S. (2018). Bayesian Naïve Bayes classifiers to text classification. *Journal of Information Science*, *44*(1), 48–59. https://doi.org/10.1177/0165551516677946

Yamada, I., Shindo, H., Takeda, H., & Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, 250–259. https://doi.org/10.18653/v1/k16-1025

Yamagiwa, H., Yokoi, S., & Shimodaira, H. (2023). *Improving word mover's distance by leveraging self-attention matrix*. 11160–11183. https://doi.org/10.18653/v1/2023.findings-emnlp.746

Yan, Y., Wang, Y., Gao, W. C., Zhang, B. W., Yang, C., & Yin, X. C. (2018). LSTM 2 : Multi-Label Ranking for Document Classification. *Neural Processing Letters*, *47*(1), 117–138. https://doi.org/10.1007/s11063-017-9636-0

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yang, X., He, X., Zhang, H., Ma, Y., Bian, J., & Wu, Y. (2020). Measurement of semantic textual similarity in clinical texts: Comparison of transformer-based models. *JMIR Medical Informatics*, *8*(11). https://doi.org/10.2196/19735

Yang, Z., Tong, K., Jin, S., Wang, S., Yang, C., & Jiang, F. (2023). CNN-Siam: multimodal siamese CNN-based deep learning approach for drug–drug interaction prediction. *BMC Bioinformatics*, *24*(1). https://doi.org/10.1186/s12859-023-05242-y

Yao, L., Pan, Z., & Ning, H. (2019). Unlabeled Short Text Similarity with LSTM Encoder. *IEEE Access*, *7*, 3430–3437. https://doi.org/10.1109/ACCESS.2018.2885698

Yaseen, H. K., & Obaid, A. M. (2020). Big data: Definition, architecture & applications. *International Journal on Informatics Visualization*, *4*(1), 45–51. https://doi.org/10.30630/joiv.4.1.292

Yih, W. tau, Toutanova, K., Platt, J. C., & Meek, C. (2011). Learning discriminative projections for text similarity measures. *CoNLL 2011 - Fifteenth Conference on Computational Natural Language Learning, Proceedings of the Conference*, 247–256.

Yogatama, D., Liu, F., & Smith, N. A. (2015). Extractive summarization by maximizing semantic volume. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 1961–1966. https://doi.org/10.18653/v1/d15-1228

Zeghdaoui, M. W., Boussaid, O., Bentayeb, F., & Joly, F. (2021). Medical-Based Text Classification Using FastText Features and CNN-LSTM Model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12923 LNCS*, 155–167. https://doi.org/10.1007/978-3-030-86472-9_15

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. *COLING 2014 - 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers*, 2335–2344.

Zhang, J., Li, Y., Tian, J., & Li, T. (2018). LSTM-CNN Hybrid Model for Text Classification. *Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018*, 1675–1680. https://doi.org/10.1109/IAEAC.2018.8577620

Zhang, R., & El-Gohary, N. (2021). A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction*, *132*. https://doi.org/10.1016/j.autcon.2021.103834

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*, *6*(1). https://doi.org/10.1038/s41597-019-0055-0

Zhao, D., Wang, J., Chu, Y., Zhang, Y., Yang, Z., & Lin, H. (2021). Improving biomedical word representation with locally linear embedding. *Neurocomputing*, *447*, 172–182. https://doi.org/10.1016/j.neucom.2021.02.071

Zhao, H., Xie, J., & Wang, H. (2022). Graph Convolutional Network Based on Multi-Head Pooling for Short Text Classification. *IEEE Access*, *10*, 11947–11956. https://doi.org/10.1109/ACCESS.2022.3146303

Zhao, X., Chumkamon, S., Duan, S., Rojas, J., & Pan, J. (2018). Collaborative

Human-Robot Motion Generation Using LSTM-RNN. *IEEE-RAS International Conference on Humanoid Robots*, *2018-Novem*, 441–446. https://doi.org/10.1109/HUMANOIDS.2018.8625068

Zhao, X., Jiang, H., Yin, J., Liu, H., Zhu, R., Mei, S., & Zhu, C. tai. (2022). Changing trends in clinical research literature on PubMed database from 1991 to 2020. *European Journal of Medical Research*, *27*(1). https://doi.org/10.1186/s40001-022-00717-9

Zheng, T., Gao, Y., Wang, F., Fan, C., Fu, X., Li, M., Zhang, Y., Zhang, S., & Ma, H. (2019). Detection of medical text semantic similarity based on convolutional neural network. *BMC Medical Informatics and Decision Making*, *19*(1). https://doi.org/10.1186/s12911-019-0880-2

Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). *A C-LSTM Neural Network for Text Classification*. http://arxiv.org/abs/1511.08630

Zong, C., Xia, R., & Zhang, J. (2021). Text Data Mining. In *Text Data Mining*. https://doi.org/10.1007/978-981-16-0100-2