TESIS

PERBANDINGAN FITUR EKSTRAKSI GLOVE DAN FASTTEXT MENGGUNAKAN METODE LONG-SHORT TERM MEMORY

Disusun oleh:

Nama : **Hannan Asrawi**
NIM : **21.55.2149**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

TESIS

PERBANDINGAN FITUR EKSTRAKSI GLOVE DAN FASTTEXT MENGGUNAKAN METODE LONG-SHORT TERM MEMORY

GLOVE AND FASTTEXT FEATURE EXTRACTION COMPARISON USING LONG-SHORT TERM MEMORY METHOD

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : **Hannan Asrawi**
NIM : **21.55.2149**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

HALAMAN PENGESAHAN

PERBANDINGAN FITUR EKSTRAKSI GLOVE DAN FASTTEXT MENGGUNAKAN METODE LONG-SHORT TERM MEMORY

GLOVE AND FASTTEXT FEATURE EXTRACTION COMPARISON USING LONG-SHORT TERM MEMORY METHOD

Dipersiapkan dan Disusun oleh

Hannan Asrawi

21.55.2149

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jum'at, 2 Februari 2024

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Februari 2014
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

PERBANDINGAN FITUR EKSTRAKSI GLOVE DAN FASTTEXT MENGGUNAKAN METODE LONG-SHORT TERM MEMORY

GLOVE AND FASTTEXT FEATURE EXTRACTION COMPARISON USING LONG-SHORT TERM MEMORY METHOD

Dipersiapkan dan Disusun oleh

Hannan Asrawi

21.55.2149

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jum'at, 2 Februari 2024

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Anggota Tim Penguji

Dr. Andi Sunyoto, M. Kom.
NIK. 190302052

Pembimbing Pendamping

Ainul Yaqin, S.Kom., M.Kom.
NIK. 190302255

Dr. Kumara Ari Yuana, S.T., M.T.
NIK. 190302575

Prof. Dr. Ema Utami, S.Si., M.Kom
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Februari 2024
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 19030210

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Hannan Asrawi
NIM : 21.55.2149
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:
Perbandingan Fitur Ekstraksi GloVe Dan FastText Menggunakan Metode Long-Short Term Memory
Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom
Dosen Pembimbing Pendamping : Ainul Yaqin, S.Kom., M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, tanggal ujian tesis
Yang Menyatakan,

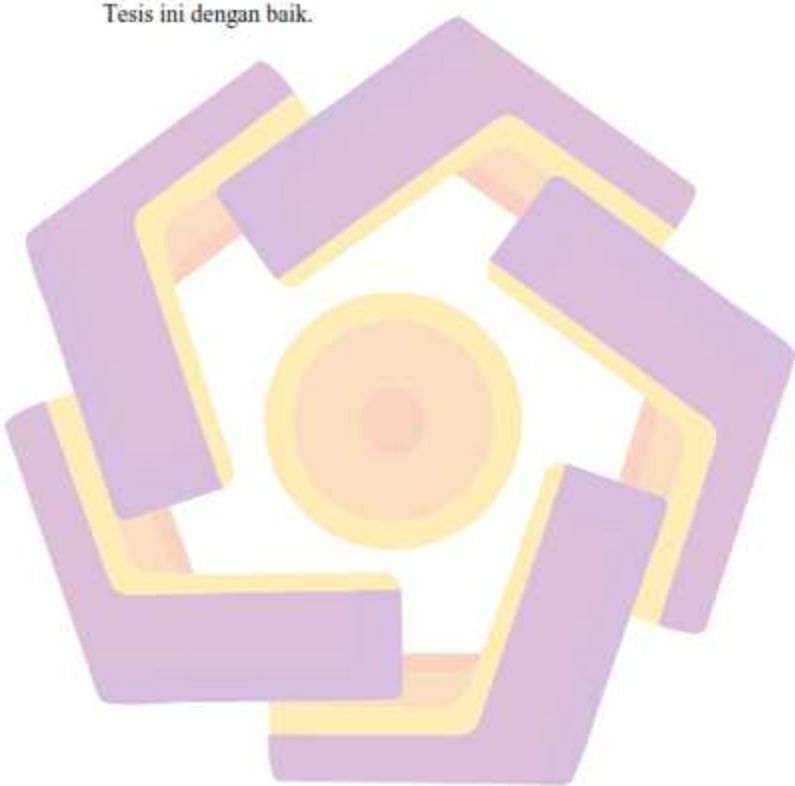
Hannan Asrawi

HALAMAN PERSEMBAHAN

Dengan rasa syukur yang mendalam, dengan telah diselesaikannya tesis ini penulis mempersembahkannya kepada:

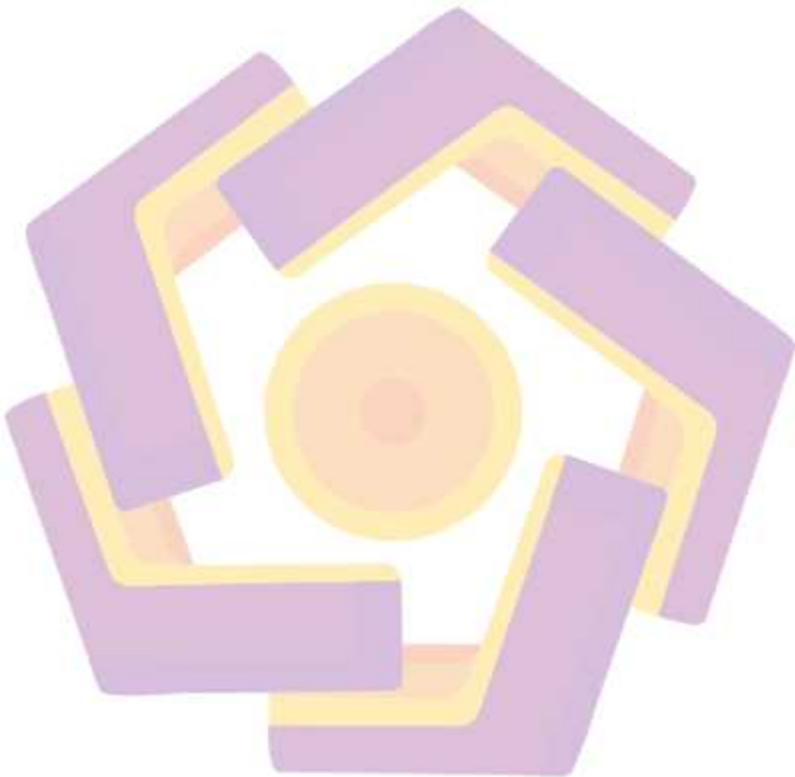
1. Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tesis ini dengan baik dan diwaktu yang tepat.
2. Shalawat dan salam semoga tercurah kepada baginda Nabi Muhammad SAW, beserta keluarga dan sahabat – sahabar beliau.
3. Kepada diri saya sendiri, yang selalu kuat dan semangat berjuang untuk tidak menyerah dan dapat menyelesaikan Tesis ini.
4. Kepada kedua orang tua saya Alm. Papa Safwan M. Ali (Al-Fatihah) dan Mama tersayang Nurhidayati, M.Ph yang memberikan dukungan dengan do'a maupun dari moril dan materil, memberikan yang terbaik dengan sepenuh hati.
5. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom., selaku Dosen Pembimbing 1 yang telah memberikan bimbingan, saran, kritik, dan motivasi kepada penulis sehingga tesis ini dapat terselesaikan.
6. Bapak Ainul Yaqin, S.Kom., M.Kom. selaku Dosen Pembimbing 2 yang telah memberikan arahan dan masukan selama penulisan tesis ini.
7. Adik adik saya Nafis, Nadin dan Asyraf yang selalu mensupport dengan cara mereka sendiri.
8. Muhammad Fauzi, S.E. saya ucapkan terimakasih karena telah membantu banyak hal dan menjadi alasan saya untuk segera menyelesaikan tesis ini.

9. Teman-teman PJJ MTI yang telah memberikan semangat, saran, masukan dan pengalaman yang tak ternilai.
10. Dan seluruh pihak yang tidak dapat saya sebutkan satu per satu, terimkasih atas segala bantuannya dan do'anya sehingga saya dapat menyelesaikan Tesis ini dengan baik.



HALAMAN MOTTO

"Usaha, doa, dan tawakkal, namun untuk hasil semua telah menjadi Ketetapan - Nya"



KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Alhamdulillahirabbil'Alamin. Segala puji bagi Allah SWT yang telah memberikan rahmat, hidayah, dan ridha-Nya, sehingga penulis dapat menyelesaikan tesis yang berjudul **"Perbandingan Fitur Ekstraksi GloVe Dan FastText Menggunakan Metode Long-Short Term Memory".**

Penulis menyadari masih terdapat banyak kekurangan selama penyusunan tesis, dan tesis ini dapat diselesaikan karena doa, dukungan, bantuan serta bimbingan dari berbagai pihak. Dalam kesempatan ini, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terimakasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas AMIKOM Yogyakarta
2. Ibu Prof. Dr. Kusrini, M.Kom., selaku Direktur Program Pascasarjana Universitas AMIKOM Yogyakarta
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom., Bapak Ainul Yaqin, S.Kom., M.Kom. selaku dosen pembimbing yang dengan sabar memberikan bimbingan, arahan, masukan, dan motivasi selama proses penulisan naskah tesis ini.
4. Segenap civitas akademika Pascasarjana, terutama seluruh dosen, yang telah memberikan ilmu dan bimbingannya.
5. Kedua orang tua juga adik adik yang tak pernah telah memberikan dukungan dan doa selama ini.

Penulis menyadari bahwa keterbatasan pengetahuan dan pengalaman penulis yang masih jauh dari harapan, untuk itu penulis sangat mengharapkan kritik, saran, dan masukan yang bersifat membangun ke arah perbaikan dan penyempurnaan tesis ini. Penulis berharap tesis ini dapat bermanfaat bagi seluruh pihak, dan semoga amal baik yang telah diberikan kepada penulis mendapat balasan dari Allah SWT. *Aamiin ya rabbal'alamin.*

Wassalamualaikum Wr.Wb.

Bireuen, 19 Januari 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Klasifikasi teks adalah salah satu bidang dari Neural Language Processing. Beberapa dari teknik dan metode dalam menyelesaikan masalah dalam NLP pada dasarnya bergantung pada kemunculan atau frekuensi kata-kata. Algoritma LSTM memiliki kelebihan dalam prediksi time series, atau prediksi data lainnya. fitur ekstraksi Glove mempelajari penyematan kata yang mengkodekan rasio probabilitas kemunculan bersama antara dua kata sebagai perbedaan vektor. FastText mempelajari representasi kata dengan mempertimbangkan informasi subword.

Penelitian ini bertujuan untuk menggunakan penerapan algoritma LSTM, dikombinasikan dengan fitur ekstraksi Glove dan FastText yang akan diklasifikasikan menggunakan algoritma LSTM menggunakan dataset 20 Newsgroup. yang mana dari kedua fitur tersebut akan dibandingkan seberapa berpengaruhnya algoritma LSTM dalam mengklasifikan teks. Kedua fitur ekstraksi dipilih karena mampu menangkap konteks di sekitar kata-kata serta implikasi semantik, sintaksis, dan sekuensial.

Evaluasi kinerja dari setiap scenario diukur menggunakan akurasi, presisi recall, f-score. Performa fitur ekstraksi FastText yang diklasifikasikan dengan LSTM lebih unggul dibandingkan dengan Glove, yaitu dengan hasil akurasi sebesar : 0.9523, presisi : 0.9651, recall: 0.9519, dan F1 Score : 0.9676. dan fitur ekstraksi Glove mendapatkan hasil nilai akurasi : 0.9562, presisi : 0.9701, recall : 0.9485 dan F1 Score : 0.9701. Perbedaan dari kedua fitur tidak begitu signifikan, yang menunjukkan kedua fitur ekstraksi memiliki kinerja yang sangat kompetitif.

Kata kunci: Klasifikasi teks, Fitur Ekstraksi, LSTM, Glove dan FastText

ABSTRACT

Among the applications of neural language processing is text classification. A number of NLP problem-solving strategies and tactics essentially depend on the frequency or appearance of words. When it comes to predicting time series or other types of data, the LSTM algorithm is advantageous. Word embedding, which stores the likelihood ratio of co-occurrence between two words as a difference vector, is learned using glove feature extraction. By taking into account subword information, FastText learns word representation.

The objective of this study is to apply the LSTM algorithm in conjunction with the Glove and FastText extraction features. A dataset consisting of 20 Newsgroups will be used for the LSTM algorithm's classification. Which of the two characteristics will be used to compare the significance of the LSTM algorithm for text classification? Due to their ability to capture word context in addition to semantic, syntactic, and sequential implications, both extraction features were selected.

Each scenario's performance is evaluated using f-score, accuracy, and precision recall. FastText feature extraction classified using LSTM performs better than Glove, with F1 Score of 0.9676, accuracy of 0.9523, precision of 0.9651, recall of 0.9519. and Glove extraction characteristics yield the following results: F1 Score: 0.9701, accuracy values: 0.9562, precision: 0.9701, recall: 0.9485. The fact that there is not much of a difference between the two characteristics indicates that both extraction features operate extremely well.

Keyword: Text Classification, Feature Extraction, LSTM, Glove, FastText

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Mempelajari sesuatu yang baru sangatlah menyenangkan, rasa penasaran yang tumbuh membuat diri ingin terus menerus larut dalam belajar, apalagi saat benar benar dipelajari dari awal, diulik sedemikian rupa sehingga menjadi sebuah pengetahuan yang baru, pengetahuan yang nantinya akan dikembangkan dengan kombinasi dari pengetahuan lainnya.

Saat ini pesatnya ilmu pengetahuan sehingga ada banyak hal yang dapat di peroleh. Salah satunya teknologi *Artificial Intelligence* atau kecerdasan buatan, yaitu simulasi proses kecerdasan manusia oleh mesin, terutama sistem komputer. Aplikasi spesifik dari AI meliputi *machine learning*, *natural language processing*, *speech recognition*, *vision* dan masih banyak lagi. Sangat menarik untuk di pelajari(Huang et al., 2018)

Pada penelitian ini dipersempit dengan mempelajari salah satu cabang dari *Artificial Intelligence* yaitu *Natural Language Processing*. Suatu cabang ilmu komputer dan lebih khusus lagi, cabang kecerdasan buatan atau AI yang berkaitan dengan kemampuan komputer untuk memahami teks dan kata-kata yang diucapkan dengan cara yang sama seperti manusia. Ini membantu mesin memproses dan memahami bahasa manusia sehingga mereka dapat secara otomatis melakukan tugas-tugas yang berulang (Kulkarni & Shivananda, 2019).

Ada banyak teknik atau metode yang dapat dipelajari dalam *Natural Language Processing* salah satunya tentang penerapan preprocessing teks dan rekayasa fitur dalam NLP, bersama dengan metode rekayasa fitur tingkat lanjut seperti *word embeddings* atau penyematan kata. Juga memahami dan mengimplementasikan konsep pencarian informasi, peringkasan teks, analisis sentimen, klasifikasi teks, pembuatan teks, dan teknik NLP tingkat lanjut lainnya yang diselesaikan dengan memanfaatkan *machine learning* and *deep learning* (K. joshi, 1991).

Beberapa dari teknik dan metode dalam menyelesaikan masalah dalam NLP pada dasarnya bergantung pada kemunculan atau frekuensi kata-kata. Namun masuk ke masalah yang lebih rumit yaitu menangkap konteks atau hubungan semantik antara kata-kata, contohnya *I am eating an apple* dan *I am using apple*. Jika diamati contoh di atas, Apple memberikan arti yang berbeda ketika digunakan dengan kata yang berbeda yang berdekatan, yaitu eating dan using.

Bagaimana cara membuat mesin memahami bahwa "Apple" dalam "Apple is a tasty fruit" adalah buah yang bisa dimakan dan bukan nama dari sebuah perusahaan? Jawaban dari pertanyaan-pertanyaan di atas terletak pada pembuatan representasi untuk kata-kata yang dapat menangkap makna, hubungan semantik, dan berbagai jenis konteks penggunaannya. Tantangan-tantangan di atas dijawab oleh *Word Embeddings*(Juwiantho et al., 2020).

Melalui metode rekayasa fitur ektraksi tingkat lanjut seperti *word embeddings* yaitu teknik pembelajaran fitur di mana kata-kata dari kosakata dipetakan ke vektor bilangan real yang menangkap hirarki kontekstual. Proses

konversi kata yang berupa karakter alfanumerik kedalam bentuk vector. Setiap kata adalah vector yang mereprentasikan sebuah titik pada space dengan dimensi tertentu(Alshari et al., 2017).

Salah satu alat yang dapat digunakan dalam text mining untuk menentukan nilai kemiripan makna kata adalah kemiripan semantik. Ada beberapa aplikasi yang dapat memanfaatkan kemiripan semantik ini. Pengukuran ini terinspirasi dari fakta bahwa saat ini komputer tidak dapat menyamai penglihatan manusia dalam menentukan seberapa mirip dua kata(Indrapurasih et al., 2018).

Ada banyak teknik fitur ekstraksi yang popular digunakan dalam NLP diantaranya yaitu Skip-Gram, Continuous Bag-of-words, Word2vec implementation, Word embedding model using Pre-trained models, Google word2vec, Stanford Glove Embeddings, FastText Facebook.

GloVe adalah model yang dikembangkan oleh Stanford, yaitu model regresi log-biner global baru untuk pembelajaran tanpa pengawasan representasi kata yang mengungguli model-model lain dalam hal analogi kata, kemiripan kata, dan tugas-tugas pengenalan entitas bernama. GloVe adalah teknologi berbasis vektor yang menggunakan statistik lokal dan global untuk mencapai tujuan dari fungsi prinsip ganda. Cara kerjanya yaitu menggunakan metode global matrix factorization, matrix yang mewakili kemunculan / ketiadaan kata dalam suatu dokumen. Bertujuan untuk mempelajari vector kata sedemikian rupa sehingga dot product kata kata tersebut sama dengan logaritma probabilitas kata kata untuk muncul bersama / probabilitas *co-occurrence* nya. (Pennington et al., 2014) (Nurdin et al., 2020).

FastText adalah pustaka sumber terbuka, yang dikembangkan oleh laboratorium Facebook AI Research. Fokus utamanya adalah mencapai solusi yang dapat diskalakan untuk tugas-tugas klasifikasi dan representasi teks sambil memproses kumpulan data yang besar dengan cepat dan akurat. FastText adalah versi modifikasi dari word2vec. FastText memperlakukan setiap kata sebagai terdiri dari n-gram. Dalam word2vec, setiap kata direpresentasikan sebagai sekumpulan kata, tetapi dalam FastText, setiap kata direpresentasikan sebagai sekumpulan karakter n-gram (Bojanowski et al., 2017).

Karakter n-gram adalah urutan yang berdekatan dari n item dari sampel karakter atau kata yang diberikan. Bisa berupa bigram, trigram, dll. Misalnya karakter trigram (n = 3) dari kata "where": <wh, whe, her, ere, re>

Dalam arsitektur FastText, mereka juga memasukkan kata itu sendiri dengan karakter n-gram. Itu berarti data masukan ke model untuk kata "di mana" adalah: <wh, whe, her, ere, re> dan <where>.

Perbedaan dari kedua representasi kata di atas yaitu perbedaan utamanya adalah Glove memperlakukan setiap kata dalam korpus seperti entitas atom dan menghasilkan vektor untuk setiap kata. Dalam pengertian ini Glove sangat mirip dengan word2vec keduanya memperlakukan kata sebagai unit terkecil untuk dilatih. Fasttext yang pada dasarnya merupakan perpanjangan dari model word2vec, memperlakukan setiap kata sebagai komposisi ngram karakter. Jadi vektor untuk sebuah kata dibuat dari jumlah karakter ini n-gram (Dharma et al., 2022) .

Dalam penelitian sebelumnya (Nurdin et al., 2020) dataset yang digunakan pada penelitian tersebut adalah 20 newsgorup (The UCI KDD Archive, 1999a) dan dataset Reuters Newswire Topic Classification (The UCI KDD Archive,1999b). Menurut temuan penelitian tersebut, CNN berkinerja baik dalam mengkategorikan teks dengan menggunakan representasi kata Word2vec, GloVe, dan FastText menggunakan F-Measure, dengan skor 0,925, 0,958, dan 0,979 untuk dataset 20 newsgroup dan 0,694, 0,688, dan 0,715 untuk dataset Reuters News. Kata-kata yang bukan bagian dari korpus (di luar kosakata) tidak dapat direpresentasikan sebagai vektor menggunakan Word2vec atau GloVe. Untuk masalah kurangnya kosakata ini, FastText dapat diandalkan. Memanfaatkan penyematan kata FastText memberikan hasil terbaik dalam percobaan. Ketiga penyematan kata tersebut memiliki kinerja yang kompetitif, meskipun perbedaan kinerjanya dapat diabaikan.

Pada penelitian (Susanty & Sukardi, 2021) Dataset yang digunakan pada penelitian ini adalah data berupa kumpulan kalimat yang telah diberi label dan diperoleh dari sumber dataset publik dari dua repositori di github, yakni repositori nlpexperiments2 dan indonesian-ner3. Masalah *Named-Entity Recognition* untuk Bahasa Indonesia telah digunakan dalam penelitian ini untuk membandingkan beberapa strategi representasi kata yang menggunakan arsitektur model BiLSTM. Karena lebih tahan terhadap *overfitting* daripada teknik *supervised* yang menggunakan lapisan embedding yang dapat dilatih, pendekatan tanpa supervisi yang menggunakan embedding yang sudah dilatih memberikan hasil yang lebih tinggi. Performa lapisan embedding yang dapat

dilatih ditingkatkan melalui pengoptimalan *hyper-parameter*, tetapi lapisan embedding yang telah dilatih sebelumnya masih lebih baik. Metode embedding dengan nilai f1 rata-rata mikro tertinggi di antara embedding yang telah dilatih yang diuji dalam penelitian ini adalah GloVe, dengan nilai 76,48. Peneliti juga menyarankan untuk mengurangi overfit model terhadap data pelatihan, penelitian selanjutnya harus menggunakan lebih banyak dataset.

Menggunakan dataset publik banyak digunakan dalam beberapa penelitian, seperti penelitian yang sudah dilakukan, dataset yang digunakan untuk percobaan ini dapat diperoleh dari GitHub. Di dalamnya terdapat sebuah korpus besar yang telah diberi tag yang sesuai, yang diambil masing-masing 1000 data dari berita yang dianotasi sebagai berita terpercaya, palsu, dan sindiran dengan total 3000 data. Karena dataset masih dalam bahasa Inggris, penerjemah online yang didukung oleh Google Translate digunakan untuk mengonversi dataset ke bahasa Indonesia (Adipradana et al., 2021).

Dalam penelitian ini akan menerapkan algoritma *Long Short-Term Memory* (LSTM). LSTM adalah algoritma yang dikembangkan dari metode *deep learning Recurrent Neural Network* (RNN). *Recurrent Neural Network* (RNN) yang kemudian dimodifikasi menjadi *Long Short-Term Memory* (LSTM) yaitu dengan menambahkan sel memori yang mampu menyimpan informasi dalam jangka waktu yang lama. Kelebihan LSTM dari pada *Recurrent Neural Network* (RNN) yaitu mampu memproses data yang relatif panjang (Ibrohim & Budi, 2018).

Dalam sebuah literatur (Sherstinsky, 2020) penulisnya menyoroti probabilitas kesalahan yang hilang, yang merupakan kekurangan serius dari RNN. LSTM memberikan solusi yang memungkinkan untuk masalah ini dengan memperkenalkan aliran kesalahan yang konstan melalui keadaan internal sel memori khusus. Dengan cara ini, LSTM mampu mengatasi masalah jeda waktu yang lama, menjembatani interval waktu lebih dari 1.000 langkah waktu. Karena itulah penelitian ini disarankan menggunkan model LSTM.

Beberapa penelitian yang tercantum di atas telah memberikan contoh serta motivasi untuk melakukan penelitian lebih lanjut saat melakukan analisis sentimen menggunakan *deep learning*. Dibandingkan dengan akurasi yang dihasilkan oleh algoritma *machine learning*, algoritma deep learning menghasilkan tingkat akurasi yang lebih tinggi (D'Sa et al., 2020).

Pada penelitian (Adipradana et al., 2021) mengemukakan hasil temuan mendukung kesimpulan yang menunjukkan bahwa penyematan kata GloVe berkinerja lebih buruk daripada penyematan kata FastText ketika digunakan dalam empat jenis pendekatan yang berbeda, termasuk LSTM, Bi-LSTM, Gated Recurrent Unit (GRU), dan Bidirectional Gated Recurrent Unit (Bi-GRU). . Hasil penelitian (Sharma et al., 2018) representasi vektor dari kata-kata menggunakan teknik tanpa pengawasan seperti Glove terbukti sangat efektif dalam menafsirkan makna dan sentimen.

Pada penelitian (Santos et al., 2017) penelitian terbaru menunjukkan bahwa CNN dapat bekerja dengan baik untuk NLP. Penelitian tersebut menyajikan penggunaan embeddings kata Facebook fastText. Hasil penelitian menunjukkan

bahwa pendekatan yang diusulkan mengungguli model dasar dan memiliki kinerja yang sama dengan model yang sudah ada. Sebagai bahan pembanding, akan diuji coba juga word embedding GloVe pada dataset yang sama.

Berdasarkan beberapa penelitian di atas, maka pada penelitian ini akan diuji metode LSTM dengan menggunakan representasi kata Glove dan sebagai bahan pembanding, akan diuji coba juga representasi kata Fasttext menggunakan dataset yang sama. Penelitian ini dilakukan untuk membandingkan kinerja dari kedua repesentasi kata tersebut.

1.2 Rumusan Masalah

Berdasarkan dari latar belakang masalah yang telah dijelaskan sebelumnya maka rumusan masalah dalam penelitian ini adalah sebagai berikut :

a. Berapa nilai performa epoch untuk akurasi, presisi, recall dan f1 score yang dihasilkan oleh model saat menggunakan masing-masing fitur ekstraksi Glove dan FastText?
b. Apakah fitur ekstraksi Glove dan FastText mempengaruhi performa dalam penerapan algoritma *Long Short-Term Memory* (LSTM)?
c. Fitur ekstraksi manakah yang menghasilkan epoch dengan akurasi tertinggi?

1.3 Batasan Masalah

Pembatasan suatu masalah digunakan untuk menghindari adanya pelebaran atau penyimpangan maupun pokok masalah agar penelitian tersebut lebih terarah

dan memudahkan dalam pembahasan sehingga tujuan penelitian akan tercapai. Berikut beberapa Batasan masalah dalam penelitian ini :

a. Data publik 20 newsgorup (The UCI KDD Archive, 1999a)
b. *Pre-processing* data dilakukan melalui teknik *remove punctuation, case folding, remove number, stopword removal, tokenization* dan *stemming*.
c. *Dataset* yang dipakai pada penelitian ini hanya berupa teks.
d. Pengambilan dan pengolahan data serta pemrosesan algoritma menggunakan Phyton.
e. *Platform* penelitian menggunakan Jupyter Notebook.
f. Metode yang digunakan adalah metode LSTM dengan beberapa kombinasi fitur ekstraksi yaitu Glove dan FastText.
g. Membandingkan hasil epoch dengan tingkat performa untuk akurasi, presisi, *recall* dan *f1 score* dari algoritma Glove + LSTM dan FastText + LSTM.

1.4 Tujuan Penelitian

Adapun tujuan penulis dalam melakukan penelitian ini adalah sebagai berikut:

a. Mengetahui nilai performa epoch untuk akurasi, presisi, recall dan f1 score yang dihasilkan oleh model saat menggunakan masing-masing fitur ekstraksi Glove dan FastText?

b. Mengetahui fitur ekstraksi Glove dan FastText mempengaruhi performa dalam penerapan algoritma *Long Short-Term Memory* (LSTM) atau tidak.

c. Mengetahui fitur ekstraksi manakah yang menghasilkan performa terbaik.

1.5 Manfaat Penelitian

Bagian ini memuat penjelasan tentang:

a. Memberikan kontribusi tentang penggunaan fitur ekstraksi menggunakan algoritma LSTM.

b. Dari hasil penelitian ini dijadikan acuan pada penelitian lebih selanjut dengan *deep learning* menggunakan penerapan fitur ekstraksi dan Glove dan FastText pada algoritma *Long Short-Term Memory* (LSTM).

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian yang telah terdahulu pernah dilakukan, relevan dan dijadikan studi literatur adalah sebagai berikut:

Dalam penelitian yang telah dilakukan sebelumnya yaitu tentang analisis sentimen mengenai vaksin COVID-19 di jelaskan bahwa dataset yang didapat dari media sosial twitter. Metode yang digunakan pada penelitian tersebut adalah Bidirectional LSTM yang akan dikombinasikan dengan word embedding yaitu fastText dan GloVe. Disebut melakukan 8 skenario pengujian untuk memeriksa performa dari word embedding, menggunakan vektor yang sudah dilatih dan yang belum dilatih. Dataset yang dikumpulkan disiapkan sebagai dataset sudah di stemming dan belum di stemming. Akurasi tertinggi dari skenario GloVe dihasilkan oleh model yang menggunakan GloVe yang telah dilatih secara mandiri dan dilatih pada dataset tanpa stemming. Akurasinya mencapai 92.5%. Di sisi lain, akurasi tertinggi dari skenario fastText dihasilkan oleh model yang menggunakan self-trained fastText dan dilatih pada dataset stemmed. Akurasinya mencapai 92.3%. Pada skenario lain yang menggunakan pre-trained embedding vector, akurasinya cukup rendah dibandingkan dengan skenario yang menggunakan self-trained embedding vector, karena data pre-trained embedding dilatih dengan menggunakan korpus Wikipedia yang berisi bahasa yang baku dan terstruktur dengan baik,

sedangkan dataset yang digunakan pada penelitian ini berasal dari Twitter yang berisi kalimat-kalimat yang tidak baku (Agustiningsih et al., 2022).

Teknik GloVe adalah sebuah model unsupervised learning pada representasi kata yang mengungguli model-model lain dalam identifikasi entitas bernama, analogi kata, dan kemiripan kata. Korpus Wikipedia Bahasa Indonesia digunakan sebagai masukan, dan implementasi penelitian yang menghasilkan nilai korelasi terbaik menggunakan parameter 100 dimensi vektor, ukuran windows, dan 50 iterasi, nilai korelasi antara skor yang dihasilkan dengan skor standar emas dari WordSim-353, SimLex-999, dan Miller Charles diperoleh dengan menggunakan korelasi pearson. Dengan nilai korelasi yang diperoleh sebesar 0.1165 untuk Miller Charles, 0.2280 untuk SimLex-999, dan 0.2849 untuk WordSim-353, maka hasil akhirnya menghasilkan nilai korelasi yang sesuai dengan gold standard (Indrapurasih et al., 2018).

Pada salah satu penelitian analisis sentiment twitter yang pernah dilakukan berbagai teknik dilakukan untuk memaksimalkan akurasi informasi. Hal ini dilakukan dalam tiga tahap untuk mendapatkan akurasi yang maksimal: pencarian dasar, penggunaan hyperparameter dan TF-IDF, dan penggunaan korpus. Akurasi analisis sentimen dapat diturunkan oleh penggunaan korpus yang tidak efisien yang disebabkan oleh kosakata yang tidak tepat. Oleh karena itu, pendekatan GloVe digunakan dalam perluasan fitur dalam korpus untuk mengatasi ketidaksesuaian kosakata dengan data yang dimiliki. Support Vector Machine dan Naive Bayes digunakan sebagai Metode Klasifikasi selain Metode GloVe untuk memperluas fitur. Perbandingan akurasi sebelum dan sesudah menyelesaikan ketiga tahap

tersebut disertakan dalam temuan penelitian. Perbandingan akurasi sebelum dan sesudah menyelesaikan ketiga tahap tersebut termasuk dalam temuan penelitian. Akurasi Nave Bayes dan Support Vector Machines meningkat dari 0.5394 dan 0.5406 menjadi 0.7786 dan 0.8323, masing-masing, dengan peningkatan sebesar 44% dan 54% (Dwi Dharma Sreya & Setiawan, 2022).

Sebuah penelitian mengemukakan bahwa pendeteksian emosi merupakan hal yang penting dalam berbagai bidang seperti pendidikan, bisnis, perekrutan karyawan. Pada penelitian ini, emosi akan dideteksi dengan teks yang berasal dari Twitter karena media sosial membuat penggunanya cenderung mengekspresikan emosi melalui postingan teks. enelitian ini akan menggunakan metode LSTM karena metode ini terbukti lebih baik dari penelitian-penelitian sebelumnya. Word embedding fast text juga akan digunakan pada penelitian ini untuk memperbaiki Word2Vec dan GloVe yang tidak dapat menangani masalah out of vocabulary (OOV). Penelitian ini menghasilkan akurasi terbaik untuk masing-masing word embedding sebagai berikut, Word2Vec menghasilkan akurasi 73,15%, GloVe menghasilkan akurasi 60,10%, fast text menghasilkan akurasi 73,15%. Kesimpulan dalam penelitian ini adalah akurasi terbaik diperoleh oleh Word2Vec dan fast text (Riza & Charibaldi, 2021).

Banyak penelitian yang dilakukan pada penyematan kata dalam domain NLP. Algoritme seperti GloVe, FastText digunakan untuk mengembangkan penyematan kata. Namun, tidak cukup banyak pekerjaan yang dilakukan pada bahasa India karena kurangnya ketersediaan sumber daya. Dataset yang diperlukan untuk menguji penyematan kata tidak tersedia untuk bahasa India. Dua algoritma

diusulkan - model Adaptive GloVe (AGM) dan model Adaptive FastText (AFM). Beradaptasi dengan proses pembuatan matriks co-occurrence dari model GloVe asli, AGM, memanfaatkan tag bagian dari ucapan, pengetahuan morfologi bahasa. AFM meningkatkan proses pembangunan kosakata dari model FastText yang asli. Pekerjaan ini melibatkan pembuatan penyematan kata untuk bahasa dengan sumber daya rendah seperti bahasa Hindi menggunakan AGM dan AFM serta pembuatan set data uji yang diperlukan untuk mengevaluasi penyematan kata. Penyematan kata AGM menunjukkan kesadaran morfologis, mencapai peningkatan akurasi sebesar 9% pada tugas analogi kata sintaksis, dibandingkan dengan model GloVe yang asli. AFM mengungguli FastText dengan akurasi 1% dalam tugas analogi kata dan peringkat 2 Spearman pada tugas kesamaan kata, memberikan kinerja yang canggih (Gaikwad & Haribhakta, 2020).

Alasan kenapa melakukan perbandingan Glove dan FastText ini adalah untuk mengetahui seberapa berpengaruhnya kedua word embedding tersebut menggunakan penerapan algoritma LSTM dan mengklasifikasikan teks. Karena LSTM memiliki kelebihan dalam prediksi time series, atau prediksi data lainnya. Seperti dalam penelitian (Hua et al., 2019) prediksi lalu lintas dan mobilitas pengguna dapat secara langsung memperoleh manfaat dari peningkatan ini karena kami memanfaatkan kumpulan data yang realistis untuk menunjukkan bahwa untuk RCLSTM, kinerja prediksi yang sebanding dengan LSTM tersedia, di mana waktu komputasi yang diperlukan jauh lebih sedikit.

2.2 Keaslian Penelitian

Tabel 2. 1 Matriks literatur review dan posisi penelitian
Perbandingan Fitur Ekstraksi Glove Dan Fasttext Menggunakan Metode Long-Short Term Memory

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (Cnn) Text Classification	Eddy Muntina Dharma, Ford Lumban Gaol, Harco Leslie Hendric Spits Warnars, Benfano Soewito, Journal of Theoretical and Applied Information Technology, 2022	untuk membandingkan akurasi dari 3 metode word embedding yaitu Word2Vec, GloVe dan FastText pada klasifikasi teks dengan menggunakan algoritma Convolutional Neural Network. Dataset yang digunakan diambil dari UCI KDD Archive, yang berisi 19.977 berita dan dikelompokkan ke dalam 20 newsgroup atau topik berita,	Hasilnya menunjukkan bahwa penyematan kata dengan metode Fast Text memiliki akurasi terbaik dalam proses klasifikasi.	Keakuratan dari ketiga penyematan kata ini bergantung pada kumpulan data yang digunakan dan domain masalah yang dibahas, oleh karena itu kumpulan data lain dan domain masalah yang harus dipecahkan dapat ditambahkan untuk penelitian di masa mendatang.	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks	Arliyanti Nurdin, Bernadus Anggo Seno Aji, Anugrayani Bustamin, Zaenal Abidin, Jurnal Teknokompak, 2020	Untuk mengklasifikasikan menggunakan Convolutional Neural Network, dengan membandingkan keefektifan alat penyematan kata seperti Word2Vec, GloVe, dan FastText. Dataset yang digunakan pada penelitian ini adalah 20 newsgorup (The UCI KDD Archive, 1999a) dan dataset Reuters Newswire Topic Classification (The UCI KDD Archive 1999b)	Kata-kata yang tidak ada dalam kosakata dan tidak ada dalam korpus tidak dapat diwakili oleh Word2vec atau GloVe sebagai vektor. Dalam hal kurangnya kosakata ini, FastText dapat diandalkan. Menggunakan metode penyematan kata FastText memberikan hasil terbaik dalam percobaan ini. Ketiga penyematan kata tersebut memiliki kinerja yang kompetitif, sebagaimana dibuktikan dengan perbedaan kinerja yang kecil.	Penggunaannya sangat bergantung pada dataset yang digunakan dan permasalahan yang ingin diselesaikan. Daftar	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection	Nabil Badria, Ferihane Kboubia, Anja Habacha Chaibi, 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022), 2022	Untuk menyaring konten yang tidak pantas ini menggunakan metode yang didasarkan pada kombinasi Glove dan FastText word embedding sebagai fitur input dan BiGRUmodel untuk mengidentifikasi ujaran kebencian dari situs web media sosial.	menunjukkan bahwa model yang kami usulkan (BiGRU Glove FT) efektif dalam mendeteksi konten yang tidak pantas. Performa sistem mencapai 84%, 87%, 93%, 90% akurasi, presisi, recall, dan f1-score secara berurutan.	mengeksplorasi lebih lanjut penggunaan arsitektur deep neural network untuk tugas pendeteksian ujaran kebencian. Juga menyelidiki penerapan teknik penyematan kata lainnya. Dan memperluas fokus ini ke dataset tambahan seperti dataset bahasa Arab.	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.
4	A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons	Flora Sakketou, Nicholas Ampazis, Knowledge-Based Systems journal, 2020	Mengadopsi teknik optimasi dari domain pembelajaran mesin dengan optimasi yang dibatasi untuk memanfaatkan pengetahuan relasional antara kata-kata.	eksperimen pada penambangan teks populer dan tugas-tugas pemrosesan bahasa alami, termasuk kemiripan kata, analogi kata, dan analisis sentimen, yang menunjukkan bahwa model yang diusulkan dapat secara signifikan.	mengeksplorasi lebih banyak jenis pengetahuan dan menerapkannya secara langsung ke dalam kerangka kerja faktorisasi matriks terkendala untuk menghasilkan embeddings berkualitas tinggi yang disesuaikan.	Mengoptimasi penggunaan Glove dengan algoritma yang di usulkan.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5.	Implementasi dan Analisis Kesamaan Semantik Antar Kata Bahasa Indonesia Menggunakan Metode GloVe	Ramanti Dwi Indrapurasih, M. Arif Bijaksana, Indra Lukmana Sardi, e-Proceeding of Engineering, 2018	Bertujuan untuk membahas tentang kemiripan semantik antar kata dalam Bahasa Indonesia dengan menggunakan metode GloVe.	Sistem yang dikembangkan dapat menggunakan pendekatan GloVe pada pasangan kata yang berasal dari standar industri untuk menghitung kemiripan semantik antar istilah. Hasil korelasi terbaik menggunakan parameter 100 dimensi vektor, ukuran windows, dan iterasi sebanyak 50, menghasilkan nilai korelasi sebesar 0.1165 pada Miller Charles, 0.2280 pada SimLex-999, dan 0.2849 pada WordSim-353. Ukuran vektor dan window adalah variabel yang mempengaruhi tingkat korespondensi semantik.	Disarankan untuk menggunkan korpus lain dalam menguji metode GloVe dan menggunakan dataset gold standard.	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6.	Hoax analyzer for Indonesian news using RNNs with fasttext and glove embeddings	Ryan Adipradana, Bagus Pradipabista Nayoga, Ryan Suryadi, Derwin Suhartono, Bulletin of Electrical Engineering and Informatics. 2021	dimaksudkan untuk membandingkan penyematan fastText dan GloVe untuk empat model deep neural networks (DNN): long short-term memory (LSTM), bidirectional long short-term memory (BI-LSTM), gated recurrent unit (GRU), dan bidirectional gated recurrent unit (BI-GRU) dalam hal nilai metrik saat mengklasifikasikan berita di antara tiga kelas: palsu, sahih, dan sindiran.	Hasil terakhir menunjukkan bahwa penyematan fastText lebih baik daripada penyematan GloVe dalam klasifikasi teks yang diawasi, bersama dengan BI-GRU + fastText yang memberikan hasil terbaik	Menyadari bahwa menerjemahkan berita berbahasa Inggris ke dalam berita berbahasa Indonesia dapat mengganggu hasil eksperimen yang sebenarnya, jadi saran adalah mencari berita berbahasa Indonesia tanpa diterjemahkan.	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.

Tabel 2.1 Lanjutan

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
7.	Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning	Ernest Lim, Esther Irawati Setiawan, Joan Santoso, Journal Of Intelligent Systems And Computation, 2019	Karena penggunaan Word2vec masih memiliki keterbatasan, karena itulah dilakukan penelitian ini dan menggunakan FastText embedding.	Sistem klasifikasi *stance* dalam bahasa Indonesia lebih tepat menggunakan FastText. Untuk klasifikasi *stance* tanpa fitur, penyematan kata fastText harus digunakan dengan vektor yang sudah dilatih sebelumnya wiki.id.bin. Dua pilihan terbaik untuk pengklasifikasi adalah CNN dan Bi-LSTM. terutama saat mengatur unit tersembunyi Bi-LSTM.	Penelitian ini dapat diperluas untuk memeriksa bagaimana kinerja dipengaruhi oleh elemen kalimat dan metadata media sosial lainnya. Pendekatan perhatian dan dampaknya terhadap akurasi klasifikasi sikap juga dapat digunakan untuk membangun pemahaman tentang makna kalimat.	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.
8.	Sentiment Analysis using Convolutional Neural Network with fastText Embeddings	Igor Santos, Nadia Nedjah, Luiza de Macedo Mourelle, IEEE LACCI 2017	Menyajikan penggunaan embeddings kata Facebook fastText yang baru sebagai representasi kata untuk melakukan tugas analisis sentimen.	Hasil penelitian menunjukkan bahwa pendekatan yang diusulkan mengungguli model dasar dan memiliki kinerja yang sama dengan model yang sudah ada.	perbaikan di masa depan adalah untuk menguji dampak dari konfigurasi hyperparameter lainnya	Membandingkan 2 metode yaitu Glove dan FastText yang diklasifikakan menggunakan algoritma LSTM.

2.3 Landasan Teori

2.4.1 Pengumpulan Data

Data menggunakan data publik dataset ini adalah kumpulan dokumen newsgroup (The UCI KDD Archive, 1999a). Koleksi 20 newsgroup telah menjadi kumpulan data yang populer untuk eksperimen dalam aplikasi teks teknik pembelajaran mesin, seperti klasifikasi teks dan pengelompokan teks. Teks dalam bahasa inggris, ada file (list.csv) yang berisi referensi ke nomor document_id dan newsgroup yang diasosiasikan dengannya. Ada juga 20 file yang berisi semua dokumen, satu dokumen per newsgroup. Newsgroup dan Document_id dapat direferensikan ke list.csv, setiap file newsgroup dalam bundel mewakili satu newsgroup. Setiap pesan dalam file adalah teks dari beberapa dokumen newsgroup yang diposting ke newsgroup tersebut.

2.4.2 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) yang di kembangkan oleh Alan Turing pada tahun 1950. Studi tentang pemodelan matematika dan komputer dari beragam fitur linguistik dan penciptaan berbagai macam sistem dikenal sebagai *Natural Language Processing* (NLP). Di antaranya adalah antarmuka multibahasa, terjemahan mesin, sistem bahasa lisan, yang menggabungkan ucapan dan bahasa alami, antarmuka kooperatif ke basis data dan basis pengetahuan yang mensimulasikan berbagai aspek interaksi manusia dengan manusia. Ilmu komputer, linguistik, logika, dan psikologi semuanya banyak dimasukkan ke dalam penelitian *Natural Language*

Processing, yang sangat interdisipliner. *Natural Language Processing* memainkan fungsi unik dalam ilmu komputer karena bertujuan untuk memodelkan bahasa secara komputasi dan banyak elemen dari disiplin ilmu ini berurusan dengan sifat-sifat linguistik dari komputasi (K. joshi, 1991).

Natural Language Processing (NLP) mengacu pada cabang ilmu komputer dan lebih khusus lagi, cabang kecerdasan buatan atau *Artificial Intelligence* yang berkaitan dengan memberi komputer kemampuan untuk memahami teks dan kata-kata yang diucapkan dengan cara yang sama seperti manusia.

Natural Language Processing adalah bentuk *Artificial Intelligence* yang memberi mesin kemampuan untuk tidak hanya membaca, tetapi untuk memahami dan menafsirkan bahasa manusia. Dengan *Natural Language Processing*, mesin dapat memahami teks tertulis atau lisan dan melakukan tugas termasuk pengenalan ucapan, analisis sentimen, dan peringkasan teks otomatis (Beysolow II, 2018).

2.4.3 *Text mining*

Text mining adalah teknik yang digunakan untuk klasifikasi dokumen, pengelompokan, ekstraksi informasi, analisis sentimen, dan pencarian informasi, dimana text mining merupakan salah satu varian dari data mining yang mencoba menemukan pola menarik dalam dataset teks yang besar (Sudiantoro et al., 2018).

Teknik penambangan teks membantu menurunkan sifat yang berbeda dari data tekstual amorf. Beberapa metode dan teknik mengarah pada penambangan teks yang terorganisir dengan baik dan akurat. Makalah ini didasarkan pada bagaimana penambangan harus dilakukan pada data tekstual. Proses penambangan teks, aplikasinya, Pengambilan informasi, Peringkasan dan berbagai metode semacam itu telah dibahas. Pendekatan yang sangat meyakinkan ditemukan karena pengamatan, karena metode mana yang diperiksa dan peningkatan metode disarankan (Tandel et al., 2019).

2.4.4 *Text Pre-processing*

Pemrosesan teks mengacu pada otomatisasi analisis teks elektronik. Hal ini memungkinkan model pembelajaran mesin untuk mendapatkan informasi terstruktur tentang teks yang akan digunakan untuk analisis, manipulasi teks, atau untuk menghasilkan teks baru (Aa Zezen Zenal Abidin, 2019).

Langkah pertama dalam processing dokumen teks adalah pra-processing. Proses ini dilakukan sebagai tokenization, yaitu penguraian kalimat menjadi bentuk yang lebih sederhana yang dikenal dengan kata atau istilah. Transform case atau mengubah huruf ke dalam bentuk huruf kecil juga dilakukan pada tahap pre-processing ini. Langkah yang sama pentingnya adalah filter stopword, yang menghapus informasi yang tidak relevan, tidak relevan, dan tidak perlu dari dokumen teks (Trianto et al., 2020).

2.4.5 Fitur ekstraksi

Ekstraksi fitur adalah konsep yang berkaitan dengan penerjemahan data mentah menjadi input yang dibutuhkan oleh algoritme pembelajaran mesin tertentu. Fitur-fitur yang diperoleh dari data mentah yang sebenarnya relevan untuk mengatasi masalah yang mendasarinya. Di sisi lain, penyematan kata pada dasarnya adalah representasi teks yang terdistribusi dalam ruang n-dimensi. Penyematan kata dalam pemrosesan bahasa alami (NLP) adalah gambar dari sebuah kata. Analisis teks memanfaatkan penyematan tersebut. Representasi ini sering kali berupa vektor bernilai nyata yang mengkodekan makna kata dengan cara yang memprediksi bahwa kata-kata yang berdekatan dalam ruang vektor akan memiliki makna yang sama (Curto et al., 2022).

Glove

GloVe atau Global Vektor adalah algoritme pembelajaran tanpa pengawasan untuk mendapatkan representasi vektor kata. Pelatihan dilakukan pada statistik kemunculan bersama kata-kata global yang dikumpulkan dari sebuah korpus, dan representasi yang dihasilkan menampilkan substruktur linear yang menarik dari ruang vektor kata (Pennington et al., 2014).

GloVe sebuah teknik yang memanfaatkan dua pendekatan yang berbeda: berbasis hitungan (misalnya PCA, analisis komponen utama) dan prediksi langsung seperti word2vec. Tidak seperti word2vec yang hanya bergantung

pada informasi dari kata-kata dengan jendela konteks lokal, algoritma GloVe juga menggabungkan informasi kemunculan bersama kata atau statistik global untuk mendapatkan hubungan semantik antara kata-kata dalam korpus. GloVe menggunakan metode faktorisasi matriks global, sebuah matriks yang merepresentasikan kemunculan atau ketiadaan kata dalam sebuah dokumen (Shanita Biere Supervisor dr Sandjai Bhulai, 2018)

Model GloVe bertujuan untuk mempelajari vector sedemikian rupa sehingga hasil kali titik (dot product) dari kata-kata tersebut sama dengan logaritma probabilitas kemunculan kata-kata tersebut secara bersamaan atau probabilitas kemunculan bersama (Dharma et al., 2022). Model Glove dapat dituliskan sebagai berikut:

$$w_i^T + \vec{w}_k + b_i + \vec{b}_k = \log(X_{ik}) \quad (1)$$

Keterangan :

w = vektor kata

$\vec{w}$ = vektor kata konteks

b_i dan b_k= bias skalar untuk kata ke-i dan konteks kata ke-k

X adalah matriks kemunculan bersama kata

X_{ik} = merepresentasikan berapa kali kata i muncul dalam konteks kata ke-k

Konteks kata sendiri merupakan kumpulan kata yang terdiri dari kata-kata yang berada sebelum dan sesudah kata i sebanyak ukuran windows yang diberikan. Kemudian setiap kata akan diberikan pembobotan dengan

cara 1/jarak. jarak dalam hal ini adalah jarak antara context word dengan posisi kata tersebut.

Kelemahan mendasar dari model ini adalah model ini memberikan bobot yang sama untuk semua kemunculan bersama, bahkan untuk kemunculan yang jarang atau bahkan tidak pernah terjadi. Meskipun kemunculan bersama yang jarang terjadi ini berisik dan memberikan lebih sedikit informasi dibandingkan dengan kemunculan bersama yang lebih umum, tergantung pada ukuran kosakata dan korpus, bahkan entri nol pun menyumbang 75-95% dari data dalam X. Untuk mengatasi masalah ini, selanjutnya menyediakan model regresi kuadrat terkecil berbobot (Pennington et al., 2014). Yang mendapatkan model tersebut dengan mengubah Persamaan (1) menjadi masalah kuadrat terkecil dan menambahkan fungsi pembobotan f (X_{ik}) ke dalam fungsi berikut:

$$J = \sum_{i,k=1}^{V} f(X_{ik})(w_i^T \widetilde{w_k} + b_i + b_k - \log(X_{ik}))^2 \qquad (2)$$

di mana V adalah ukuran kosakata. Fungsi pembobotan harus memenuhi sifat-sifat berikut:

1. f (0) = 0. Jika f dipandang sebagai sebuah fungsi kontinu, fungsi tersebut harus lenyap ketika $x \rightarrow 0$ dengan cepat cukup cepat sehingga $\lim_{x \rightarrow 0} f(x) \log^2 x$ adalah terbatas.
2. $f(x)$ haruslah tidak menurun sehingga kejadian kejadian yang jarang terjadi tidak diberi bobot berlebih.

3. *f (x)* harus relatif kecil untuk nilai x yang besar, sehingga kejadian bersama yang sering terjadi tidak tertimbang secara berlebihan.

Tentu saja banyak sekali fungsi yang memenuhi sifat-sifat ini ini, tetapi satu kelas fungsi yang ditemukan bekerja dengan baik dapat diparameterkan sebagai berikut:

$$f(X_{ik}) = \begin{cases} (\frac{X_{ik}}{xmax})^{\alpha}; if\ X_{ik} < xmax \\ 1; lainnya \end{cases} \quad (3)$$

FastText

FastText dapat didefinisikan sebagai metode penyematan kata yang merupakan bagian dari pengembangan word2vec (Lim et al., 2019), Dikembangkan oleh tim peneliti AI Facebook, model ini terinspirasi oleh penelitian yang dilakukan oleh Mikolov dkk (Mikolov et al., 2013). dan berhasil menunjukkan bahwa model mereka dapat melatih 1 miliar kata dalam waktu 10 menit, dengan kualitas hasil yang lebih baik dibandingkan dengan model lainnya (Bojanowski et al., 2017). Arsitektur model FastText mirip dengan arsitektur CBOW pada Word2Vec, namun memiliki struktur hirarki dan merepresentasikan kata dalam bentuk vektor yang padat. Selain itu, terdapat lapisan tersembunyi di antara lapisan input dan lapisan output.

Pendekatan fastText didasarkan pada model skip-gram, di mana setiap kata direpresentasikan sebagai sekumpulan karakter n-gram(Salur & Aydin, 2020). Sebuah representasi vektor diasosiasikan dengan setiap karakter n-gram; kata-kata direpresentasikan sebagai jumlah dari representasi ini. Representasi kata dipelajari dengan mempertimbangkan jendela besar kata-

kata konteks kiri dan kanan, tidak seperti embedding Mikolov, fastText dapat memberikan embedding untuk kata yang salah eja, kata yang jarang digunakan, atau kata yang tidak ada di dalam korpus pelatihan, karena fastText menggunakan tokenisasi kata n-gram karakter(Adipradana et al., 2021).

Setiap kata sebagai terdiri dari n-gram. Katakanlah nilai n adalah 3 untuk kata 'India', kita memiliki '<in', 'ind', 'ndi', 'di>' sebagai representasi n-gram. Dan untuk kata 'India' kita dapat menyimpulkan seluruh vektor sebagai jumlah dari representasi vektor semua karakter n-gram (Di sini diasumsikan bahwa nilai *hyperparameter* [minn] dan [maxn] adalah 3, di mana 'minn' dan 'maxn' adalah ngram terkecil dan terbesar). Simbol '<' dan '>' adalah simbol khusus dan ditambahkan untuk menunjukkan awal dan akhir token (P.S. <her> dan 'her' tidak sama).

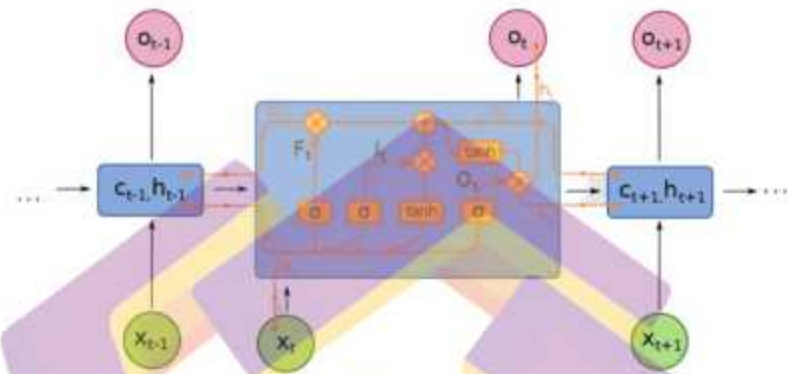
Fasttext dapat menghasilkan penyematan untuk kata-kata yang tidak muncul dalam korpus pelatihan. Hal ini dapat dilakukan dengan menambahkan karakter n-gram dari semua representasi n-gram. Sebagai contoh, katakanlah ada kata 'umumnya' di dataset pengujian, tetapi tidak memiliki representasi apa pun di set pelatihan. Tetapi set pelatihan memiliki representasi vektor dari semua n-gramnya(Badri et al., 2022). Jadi kita bisa merata-ratakan representasi vektor dari semua n-grams.word penyusunnya. Di sisi lain, untuk kata acak 'fgghoio' kita bisa mendapatkan representasi dari rata-rata semua karakter ngram (yaitu 'f'+ 'g' + 'g' + 'h' + 'o' + 'i' + 'o' di sini kita harus menjaga hiperparameter minn sebagai 1).

2.4.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah singkatan dari jaringan memori jangka pendek yang digunakan dalam bidang *Deep Learning*. Ini adalah berbagai jaringan saraf berulang *Recurrent neural networks* (RNN) yang mampu mempelajari ketergantungan jangka panjang, terutama dalam masalah prediksi urutan. LSTM memiliki koneksi umpan balik, yaitu mampu memproses seluruh urutan data, selain dari titik data tunggal seperti gambar. Ini menemukan aplikasi dalam pengenalan ucapan, terjemahan mesin, dll. LSTM adalah jenis *Recurrent neural networks* (RNN) khusus, yang menunjukkan kinerja luar biasa pada berbagai macam masalah (Hochreiter, Sepp; Schmidhuber, 1997) (Sherstinsky, 2020).

Dalam tantangan prediksi urutan, metode LSTM adalah jenis *Recurrent neural networks* (RNN) yang dapat mempelajari ketergantungan pesanan. Output dari langkah sebelumnya digunakan sebagai input pada langkah saat ini di *Recurrent neural networks* (RNN). Hochreiter & Schmidhuber menciptakan LSTM. Ini mengatasi masalah ketergantungan jangka panjang *Recurrent neural networks* (RNN), di mana *Recurrent neural networks* (RNN) tidak dapat memprediksi kata yang disimpan dalam memori jangka panjang tetapi dapat membuat prediksi yang lebih akurat berdasarkan data saat ini. *Recurrent neural networks* (RNN) tidak memberikan kinerja yang efisien karena panjang celah meningkat. LSTM dapat menyimpan

informasi untuk waktu yang lama secara default. Ini digunakan untuk pemrosesan, prediksi, dan klasifikasi data deret waktu.



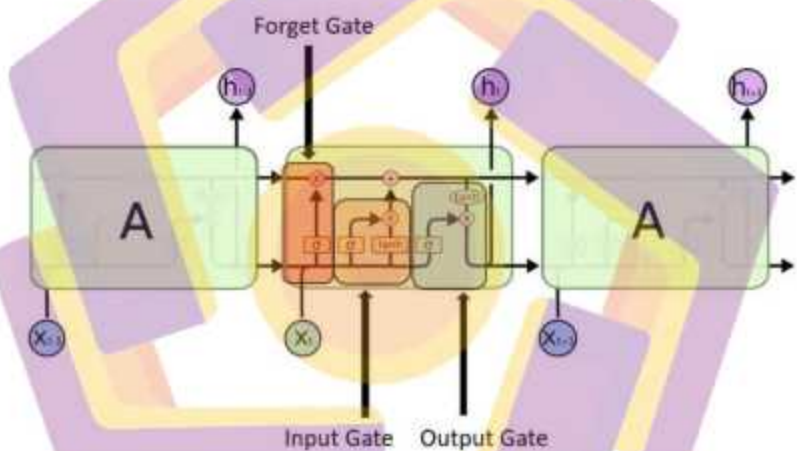
Gambar 1. Algoritma *Long Short-Term Memory* (LSTM)

LSTM memiliki koneksi umpan balik, tidak seperti jaringan saraf feed-forward konvensional. Itu tidak hanya dapat menangani titik data tunggal (seperti foto) tetapi juga aliran data lengkap (seperti ucapan atau video). LSTM dapat digunakan untuk tugas-tugas seperti tidak tersegmentasi, pengenalan tulisan tangan tertaut, atau pengenalan suara.

LSTM sebagian besar digunakan untuk mempelajari, memproses, dan mengklasifikasikan data berurutan karena jaringan ini dapat mempelajari ketergantungan jangka panjang antara langkah waktu data. LSTM umum meliputi analisis sentimen, pemodelan bahasa, pengenalan ucapan, dan analisis video.

A. Struktur *Long Short-Term Memory* (LSTM)

LSTM terdiri dari empat jaringan saraf dan banyak blok memori yang dikenal sebagai sel dalam struktur rantai. Unit LSTM konvensional terdiri dari sel, gerbang input, gerbang keluaran, dan gerbang lupa. Aliran informasi ke dalam dan ke luar sel dikendalikan oleh tiga gerbang, dan sel mengingat nilai dalam interval waktu yang sewenang-wenang. Algoritma LSTM diadaptasi dengan baik untuk mengkategorikan, menganalisis, dan memprediksi rangkaian waktu dengan durasi yang tidak pasti.



Gambar 2. Struktur Algoritma *Long Short-Term Memory* (LSTM)

Sel menyimpan informasi, sedangkan gerbang memanipulasi memori. Ada tiga pintu masuk:

- Input Gate: Ini menentukan nilai input mana yang harus digunakan untuk mengubah memori. Fungsi sigmoid menentukan apakah akan mengizinkan nilai 0 atau 1 lewat. Dan fungsi tanh memberi

bobot pada data yang diberikan, menentukan kepentingannya pada skala -1 hingga 1.

$$i_t = \sigma(W_i.[h_t - 1, x_t] + bi)$$

$$C_t = tanh\ (W_c.[h_t - 1, x_t] + bC)$$

Keterangan :

i_t: input gate

W_i : weights input

σ : sigmoid activation function

h_t - 1: output cell previously

x_t : input cell

bi : bias input gate

C_t: candidate

$tanh$: tanh act/ivation function

W_c : weights candidate

bC : bias candidate

- Forget Gate: Ia menemukan detail yang harus dihapus dari blok. Itu ditentukan oleh fungsi sigmoid. Untuk setiap angka dalam keadaan sel Ct-1, ini melihat keadaan sebelumnya (ht-1) dan input konten (Xt) dan menghasilkan angka antara 0 dan 1.

$$f_t = \sigma(W_f.[h_t - 1, x_t] + bf)$$

Keterangan :

f_t : forget gate

σ : sigmoid activation function

W_f: weights forget gate

$h_t - 1$: output cell previously

x_t: input cell

bf : bias forget gate

- Output Gate: Input dan memori blok digunakan untuk menentukan output. Fungsi sigmoid menentukan apakah akan mengizinkan nilai 0 atau 1 lewat. Dan fungsi tanh menentukan nilai mana yang diizinkan melewati 0, 1. Dan fungsi tanh memberikan bobot pada nilai yang diberikan, menentukan relevansinya pada skala -1 hingga 1 dan mengalikannya dengan output sigmoid.

$$O_t = \sigma(W_O.[h_t - 1, x_t] + bO)$$

Keterangan :

O_t : Output gate

σ : sigmoid activation function

W_O: weights output gate

$h_t - 1$: output cell previously

x_t: input cell

bO : bias output gate

- Hidden layer: berpengaruh untuk nilai di proses selanjutnya, nilai dari layer ini berasal dari nilai output yang dikalikan dengan nilai

dari cell state atau memory cell yang telah diaktivasi dengan fungsi tangen.

$$h_t = O_t * \tanh(C_t)$$

Keterangan :

h_t : hidden layer

O_t : Output gate

$tanh$: tanh act/ivation function

C_t: candidate

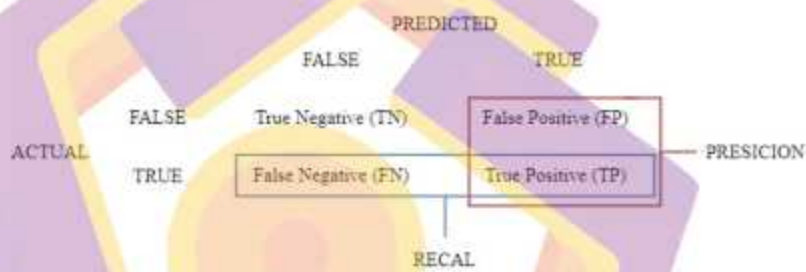
2.4.7 Model Evaluasi

Keras, sebuah pustaka neural network yang ramah pengguna yang ditulis dalam bahasa Python, adalah pilihan populer di antara para ilmuwan data karena kesederhanaan dan kemudahan penggunaannya. Salah satu fungsi yang paling sering digunakan dalam Keras adalah model.evaluate().

Evaluasi adalah proses selama pengembangan model untuk memeriksa apakah model tersebut paling cocok untuk masalah yang diberikan dan data yang sesuai. Model Keras menyediakan sebuah fungsi, evaluasi yang melakukan evaluasi model. Fungsi ini memiliki tiga argumen utama yaitu data uji, label data uji, dan verbose - true or false.

Keras v2, API neural mnetwork tingkat tinggi, mampu berjalan di atas TensorFlow, CNTK, atau Theano. API ini memungkinkan pembuatan prototipe yang mudah dan cepat serta mendukung jaringan konvolusional dan jaringan berulang, serta kombinasi keduanya.

Dalam Keras v2, Anda dapat menghitung presisi, recall, dan skor F1 menggunakan modul metrik. Namun, sangat penting untuk dicatat bahwa Keras tidak memiliki metrik-metrik ini di dalamnya setelah versi 2.0 karena sifat "global" dari metrik-metrik ini. Kunci dari model yang sukses tidak hanya pada arsitekturnya tetapi juga pada seberapa akurat Anda dapat mengukur kinerjanya.



Gambar 3. Metrik Pengukuran

Seperti pada gambar di atas terdiri dari empat karakteristik dasar (angka) yang digunakan untuk mendefinisikan metrik pengukuran pengklasifikasi. Keempat angka tersebut adalah:

TP True Positif: Nilai prediksi yang diprediksi dengan benar sebagai positif yang sebenarnya

FP False Positif: Nilai yang diprediksi salah memprediksi positif yang sebenarnya. yaitu, nilai negatif yang diprediksi sebagai positif

FN False Negatif: Nilai positif diprediksi sebagai negative

TN True Negatif: Nilai prediksi diprediksi dengan benar sebagai negatif yang sebenarnya

Metrik kinerja dari sebuah algoritma adalah akurasi, presisi, recall, dan skor F1, yang dihitung berdasarkan TP, TN, FP, dan FN yang telah disebutkan di atas.

Akurasi algoritma direpresentasikan sebagai:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Presisi Metrik presisi menunjukkan keakuratan kelas positif. Metrik ini mengukur seberapa besar kemungkinan prediksi kelas positif benar.

$$Precision = \frac{TP}{TP + FP}$$

Recall menghitung rasio kelas positif yang terdeteksi dengan benar. Metrik ini menunjukkan seberapa baik model mengenali kelas positif.

$$Recall = \frac{TP}{TP + FN}$$

F score: F1 score adalah skor rata-rata tertimbang dari nilai positif (recall) dan presisi.

$$F1\ Score = \frac{2 * precision * recall}{precision + recall}$$

BAB III

METODE PENELITIAN

3.1 Jenis, Sifat, dan Pendekatan Penelitian

Jenis penelitian yang digunakan peneliti adalah penelitian kuantitatif, dimana peneliti melakukan perhitungan matematis untuk mendapatkan hasil yang diinginkan. Dimana penelitian ini melakukan perbandingan pada pengujian tingkat akurasi yang tertinggi menggunakan fitur ekstraksi Glove dan FastText kemudian menggunakan Algoritma LSTM dengan jumlah dataset yang sama. Pengujian ini dilakukan untuk mengetahui metode yang lebih akurat dan tepat.

Penelitian ini cukup deskriptif karena menggambarkan objek tertentu yang akan dievaluasi dan menyajikan hasil dari banyak percobaan yang dilakukan pada kumpulan data yang ada sehingga metode terbaik dengan akurasi, presisi, recall, dan skor F1 terbaik.

Penelitian ini menggunakan pendekatan kuantitatif, hasilnya berupa grafik yang menunjukkan hasil eksperimen yang membandingkan penggabungan fitur Glove dan FastText ke dalam algoritma Long Short-Term Memory (LSTM).

3.2 Metode Pengumpulan Data

Data menggunakan data publik dari kaggle.com, Dataset ini adalah kumpulan dokumen newsgroup (The UCI KDD Archive, 1999a). Koleksi 20 newsgroup telah menjadi kumpulan data yang populer untuk eksperimen dalam aplikasi teks teknik pembelajaran mesin, seperti klasifikasi teks dan pengelompokan teks. Teks dalam

bahasa inggris, ada file (list.csv) yang berisi referensi ke nomor document_id dan newsgroup yang diasosiasikan dengannya.

3.3 Metode Analisis Data

Analisis data yang dilakukan adalah melakukan pre-processing teks, pada tahap ini dilakukan case folding, tokenization, filtering atau penghapusan dan penghilangan stopwords terhadap semua data. Melakukan extraksi fitur yaitu word embedding menggunakan GloVe dan FastText terhadap data hasil pre-processing.

Melakukan split data untuk menguji dan melatih data yang diekstraksi. Selanjutnya dilakukan pelatihan algoritma pada semua data pelatihan algoritma pengujian. Melakukan klasifikasi menggunkan metode LSTM dari data yang dimiliki. Juga melakukan percobaan akurasi, presisi, recall dan F1 scorenya dari hasil klasifikasi. Selanjutnya hasil analisis tersebut akan diadikan acuan atau pedoman dalam menentukan hasil atau analisis sentimen pada penelitian ini.

3.4 Alur Penelitian

Alur penelitian yang digunakan pada penelitian ini sebagai berikut:

3.4.1 Data Collection

Data menggunakan data publik dari kaggle.com, Dataset ini adalah kumpulan dokumen newsgroup. Koleksi 20 newsgroup terdiri dari sekitar 18.000 posting newsgroup pada 20 topik yang dibagi menjadi dua subset: satu untuk pelatihan satu lagi untuk pengujian. Pembagian antara set pelatihan dan pengujian didasarkan pada pesan yang diposting sebelum dan

sesudah tanggal tertentu, dataset ini telah menjadi kumpulan data yang populer untuk eksperimen dalam aplikasi teks teknik pembelajaran mesin, seperti klasifikasi teks dan pengelompokan teks.

Data ini menampilkan 20 topik berita yaitu :

1. alt.atheism.txt
2. comp.graphics.txt
3. comp.os.ms-windows.misc.txt
4. comp.sys.ibm.pc.hardware.txt
5. comp.sys.mac.hardware.txt
6. comp.windows.x.txt
7. misc.forsale.txt
8. rec.autos.txt
9. rec.motorcycles.txt
10. rec.sport.baseball.txt
11. rec.sport.hockey.txt
12. sci.crypt.txt
13. sci.electronics.txt
14. sci.med.txt
15. sci.space.txt
16. soc.religion.christian.txt
17. talk.politics.guns.txt
18. talk.politics.mideast.txt
19. talk.politics.misc.txt

20. talk.religion.misc.txt

3.4.1 Data Training dan Testing

Dataset yang didapat sudah terbagi menjadi 2 subset yaitu subset train dan subset test. Selanjutnya data yang telah melalui proses word embedding Glove dan FastText akan diklasifikasikan menggunakan algoritma LSTM. Pada prosesn training akan dibangun model LSTM yang akan melibatkan 3 lapisan, diantaranya adalah *hidden layer* (lapisan *embedding*), *input layer* (lapisan LSTM), dan *output layer* (lapisan *dense*). Dari hasil pengujian didapatkan nilai akurasi, presisi, recall, dan F1 untuk setiap jenis data latih.

3.4.2 Data Pre-processing

Sebelum melanjutkan pada proses selanjutnya, data di pre-processing terlebih dahulu. Yaitu menggunakan teknik *data cleaning, stopword removal, lower casing, dan tokenization.*

3.4.3 Fitur Ekstraksi

Fitur ekstraksi atau *Word embedding* yang digunakan adalah Glove dan FastText. Glove dilakukan pada statistik kemunculan bersama kata-kata global yang dikumpulkan dari sebuah korpus, dan representasi yang dihasilkan menampilkan substruktur linear yang menarik dari ruang vektor kata. Pada penelitian ini dataset glove yang digunakan adalah korpus gabungan Wikipedia 2014 + Gigaword Edisi ke-5 (6B token, 400K

kosakata). Semua token dalam huruf kecil. Dataset vektor kata yang digunakan adalah 200 dimensi.

FastText kemampuannya menghasilkan vektor untuk kata apa pun, bahkan kata yang dibuat-buat. Memang, vektor kata fastText dibuat dari vektor substring karakter yang terkandung di dalamnya. Vektor-vektor yang sudah dilatih ini berisi 1 juta vektor kata yang dipelajari menggunakan Wikipedia 2017, korpus basis web UMBC, dan kumpulan data berita statmt.org. vector yang digunakan adalah 300 dimensi.

3.4.4 Model LSTM

Membangun model LSTM untuk proses klasifikasi yang akan dijalankan. Lapisan LSTM jaringan ini terdiri dari lapisan tersembunyi dengan empat blok LSTM atau neuron, lapisan yang terlihat dengan satu input, dan lapisan output yang memprediksi satu nilai. Sel memori dikontrol oleh tiga gerbang: gerbang masukan, gerbang lupa, dan gerbang keluaran.

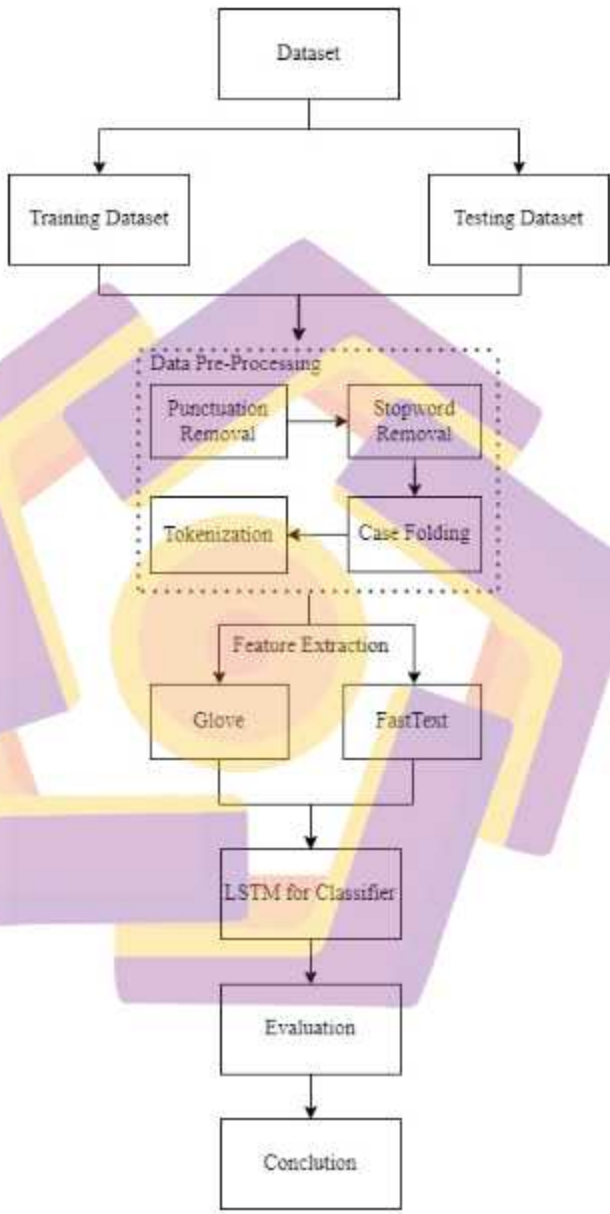
3.4.5 Model Evaluasi

Model Keras menyediakan sebuah fungsi, evaluasi yang melakukan evaluasi model. Fungsi ini memiliki tiga argumen utama yaitu data uji, label data uji, dan verbose - true or false. Model. Evaluasi memprediksi output untuk input yang diberikan dan kemudian menghitung fungsi metrik yang ditentukan dalam model. kompilasi dan berdasarkan y_true dan y_pred dan mengembalikan nilai metrik yang dihitung sebagai output.

Sebuah proses selama pengembangan model untuk memeriksa apakah model tersebut paling cocok untuk masalah yang diberikan dan data yang

sesuai. Model Keras menyediakan sebuah fungsi, evaluasi yang melakukan evaluasi model.

Analisis dan temuan dilakukan setelah semua skenario diimplementasikan dan mendapatkan kesimpulannya. Seluruh proses penelitian diilustrasikan dalam diagram di bawah ini.



Gambar 4. Alur Penelitian

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Hasil penelitian membandingkan fitur ekstraksi Glove dan FastText yang selanjutnya diklasifikasikan menggunakan metode Long-Short Term Memory. Dataset yang digunakan adalah data public yaitu koleksi 20 newsgroup telah menjadi kumpulan data berbahasa inggris yang populer untuk eksperimen dalam aplikasi teks teknik pembelajaran mesin, seperti klasifikasi teks dan pengelompokan teks. Hasil performa dari kedua fitur ektraksi yang telah di klasifikasikan menggunakan model Long-Short Term Memory dapan dilihat dari nilai akurasi, presisi, recall, f1-score. Kemudian kedua fitur yang digunakan dalam penelitian ini akan dilihat dan dibandingkan hasil kinerjanya.

4.1 Dataset

Penelitian ini menggunakan platform Jupyter Notebook dengan Bahasa pemograman Python. Dataset digunakan dalam penelitian ini, yaitu data 20 Newsgroup sekitar 18.846 dokumen newsgroup, yang dipartisi (hampir) merata di 20 newsgroup yang berbeda. Sehingga setiap topik berita terdiri dari rata-rata 1.000 berita yang diambil dari The UCI KDD Archive. Dokumen ini ditulis dalam bahasa Inggris, dan sebuah berkas bernama list.csv berisi referensi ke nomor document_id dan newsgroup yang menyertainya. Satu bagian dari posting ini ditujukan untuk pengujian (atau evaluasi kinerja), Data ini dibagi menjadi 11.314 data training. Sedangkan bagian lainnya ditujukan untuk pelatihan atau pengembangan yang berisi 7.532 data training. Berdasarkan pesan yang diposting sebelum dan sesudah tanggal tertentu, set pelatihan dan pengujian dibagi. Beberapa penelitian

menggunkan dataset ini untuk mengklasifikasikan teks dengan berbagai model klasifikasi (Chen & Dai, 2021) (Wan et al., 2019) (Dharma et al., 2022).

Untuk sampel dokumen dari dataset yang digunakan dapat dilihat pada gambar berikut.

4.2 Data Training dan Testing

Pada penelitian ini menggunakan fungsi dari fungsi sklearn.datasets dalam mengakses data dari 20 newsgroup. Dataset 20 newsgroup terdiri dari sekitar 18.846 dokumen posting tentang 20 topik. Posting ini dibagi menjadi dua subset: satu untuk pengujian (atau evaluasi kinerja) dan yang lainnya untuk pelatihan atau pengembangan. Pemisahan antara rangkaian pelatihan dan pengujian didasarkan pada pesan yang diposting sebelum dan sesudah tanggal tertentu (Sklearn.datasets, n.d.).

Pada gambar 6 adalah data dengan menggunakan subset all terdapat jumlah data atau nilai X sebanyak 18.846 dokumen dengan jumlah target atau nilai y sebanyak 18.846 dokumen.

```
dataset = fetch_20newsgroups(subset='all')

X = pd.Series(dataset['data'])
y = pd.Series(dataset['target'])
X.shape, y.shape
```

```
((18846,), (18846,))
```

Gambar 5. Nilai X dan y pada subset all

Seperti yang sudah dijelaskan sebelumnya penelitian ini menggunkan fungsi sklearn.datasets.fetch_20newsgroups jadi data train dan test langsung

diakses dari fungsi tersebut. Dalam gambar 7 menjelaskan bahwa jumlah nilai X atau data dan nilai y atau target pada subset train adalah 11.314 dokumen.

```
dataset = fetch_20newsgroups(subset='train')

X = pd.Series(dataset['data'])
y = pd.Series(dataset['target'])
X.shape, y.shape
```

```
((11314,), (11314,))
```

Gambar 6. Nilai X dan y pada subset train atau pengujian

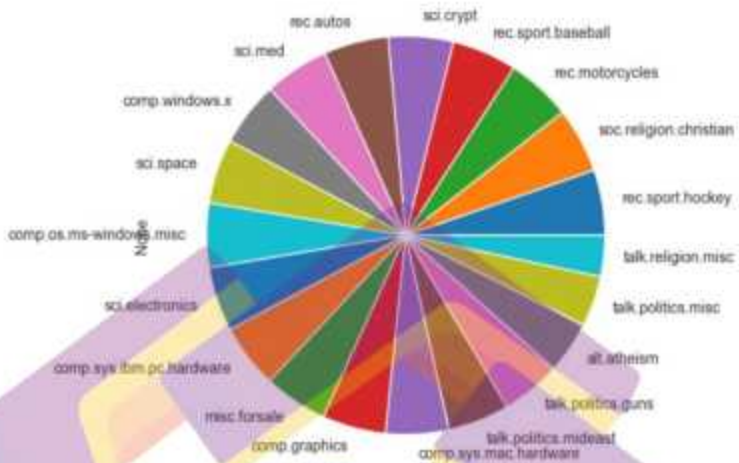
Jumlah nilai X atau data dan nilai y atau target untuk subset test dapat dilihat pada gambar 7 sebanyak 7.532 dokumen.

```
dataset = fetch_20newsgroups(subset='test')

X = pd.Series(dataset['data'])
y = pd.Series(dataset['target'])
X.shape, y.shape
```

```
((7532,), (7532,))
```

Gambar 7. Nilai X dan y pada subset test



Gambar 8. Visualisasi Class Target

4.3 Dataset Preprocessing

Pada tahap data preprocessing langkah langkah yang digunakana adalah cleansing data. Cleansing data adalah pada proses ini menghapus duplikat kata, mengecek data yg hilang, case folding menghapus semua tanda baca, huruf kecil semua karakter, dan mengonversi semua angka pada halaman menjadi huruf kecil [!"#$%&'()*+,-./:;=>?@[] (Kulkarni & Shivananda, 2021). Data Cleaning bukan hanya tentang menghapus informasi untuk memberi ruang bagi data baru, melainkan menemukan cara untuk memaksimalkan akurasi kumpulan data tanpa harus menghapus informasi (Yin et al., 2019). Selanjutnya tokenizing, yaitu memecah teks menjadi potongan-potongan kecil yang disebut token untuk analisis lebih lanjut, dan penghilangan kata penghenti, yang menghilangkan kata-kata yang sering digunakan namun

dianggap tidak bernilai (Buntoro, 2017). Proses ini melibatkan penguraian dokumen teks menjadi unit-unit yang lebih kecil yang disebut token, yang dapat berupa kata, frasa, atau karakter individual. Token-token ini kemudian dapat digunakan sebagai representasi vektor dari dokumen teks untuk melakukan berbagai tugas NLP seperti analisis sentimen, pengenalan entitas bernama, dan klasifikasi teks (Rosid et al., 2020). Tujuan tokenisasi adalah untuk menemukan representasi kecil dari teks yang paling masuk akal untuk model pembelajaran mesin (Hassani et al., 2021).

4.4 Fitur Ekstraksi

Pada penelitian ini menggunakan fitur ekstraksi Glove dan Fasttext.

4.4.1 Glove

GloVe adalah algoritme pembelajaran tanpa pengawasan untuk mendapatkan representasi vektor kata. Pelatihan dilakukan pada statistik kemunculan bersama kata-kata global yang dikumpulkan dari sebuah korpus, dan representasi yang dihasilkan menampilkan substruktur linear yang menarik dari ruang vektor kata(Gupta et al., 2021).

Dataset ini berisi vektor kata bahasa Inggris yang telah dilatih sebelumnya pada gabungan korpora Wikipedia 2014 + Gigaword Edisi ke-5 (6B token, 400 ribu kosakata). Semua token dalam huruf kecil. Dataset ini berisi vektor kata yang sudah dilatih 50 dimensi, 100 dimensi, dan 200 dimensi. Untuk vektor kata 300 dimensi dan informasi tambahan, silakan lihat situs web

proyek(Sakketou & Ampazis, 2020). Yang digunakan dalam penelitian ini yaitu vector dengan kata 300 dimensi.

4.4.2 FastText

FastText adalah sebuah pustaka untuk pembelajaran representasi kata dan klasifikasi kalimat yang efisien. Salah satu fitur utama representasi kata fastText adalah kemampuannya untuk menghasilkan vektor untuk kata apa pun, bahkan kata yang dibuat-buat. Memang, vektor kata fastText dibuat dari vektor substring karakter yang terkandung di dalamnya. Hal ini memungkinkan untuk membuat vektor bahkan untuk kata-kata yang salah eja atau gabungan kata (Fudholi et al., 2022).

Vektor-vektor yang sudah dilatih ini berisi 1 juta vektor kata yang dipelajari menggunakan Wikipedia 2017, korpus basis web UMBC, dan kumpulan data berita statmt.org. Secara keseluruhan, file ini berisi 16B token (D'Sa et al., 2020).

Baris pertama file berisi jumlah kata dalam kosakata dan ukuran vektor. Setiap baris berisi sebuah kata yang diikuti dengan vektornya, seperti dalam format teks fastText default. Setiap nilai dipisahkan dengan spasi. Kata-kata diurutkan berdasarkan frekuensi menurun (Bojanowski et al., 2017).

4.5 Model LSTM untuk Klasifiklasi

Selanjutnya adalah membangun model LSTM untuk mengklasifikasikan. Lapisan LSTM jaringan ini terdiri dari lapisan tersembunyi dengan empat blok LSTM atau neuron, lapisan yang terlihat dengan satu input, dan lapisan output

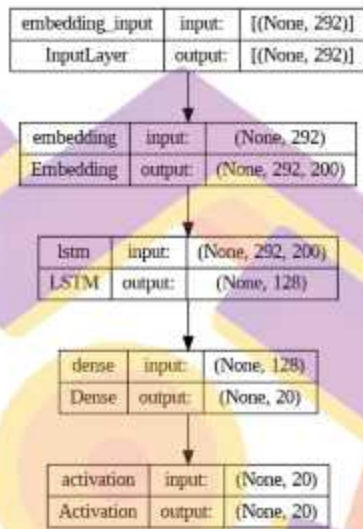
yang memprediksi satu nilai. Sel memori dikontrol oleh tiga gerbang: gerbang masukan, gerbang lupa, dan gerbang keluaran. Gerbang-gerbang ini memutuskan informasi apa yang akan ditambahkan, dihapus, dan dikeluarkan dari sel memori (Sherstinsky, 2020).

Gerbang masukan mengontrol informasi apa yang ditambahkan ke sel memori. Gerbang lupa mengontrol informasi apa yang dihapus dari sel memori. Dan gerbang keluaran mengontrol informasi apa yang dikeluarkan dari sel memori. Hal ini memungkinkan jaringan LSTM untuk secara selektif mempertahankan atau membuang informasi saat mengalir melalui jaringan, yang memungkinkan mereka untuk mempelajari ketergantungan jangka panjang(Hua et al., 2019).

Blok-blok LSTM menggunakan fungsi aktivasi sigmoid default (Chopra et al., 2019). LSTM selalu memiliki array 3D sebagai lapisan inputnya. Bergantung pada argumennya, LSTM dapat menghasilkan array 2D atau array 3D, namun dalam hal ini outputnya adalah array 2D.

Ilustrasi dari ringkasan model Long Short-Term Memory seperti yang digambarkan pada gambar 9. Keluaran dari lapisan padat berikutnya adalah hasil kali titik dari matriks bobot, yang juga dikenal sebagai kernel, dan tensor input. Fungsi softmax juga digunakan oleh lapisan aktivasi untuk mengubah vektor nilai menjadi distribusi probabilitas (Muhammad et al., 2021). Komponen vektor output, yang bervariasi dari 0 hingga 1, dijumlahkan hingga 1. Setiap vektor ditangani secara terpisah. Argumen axis memberi tahu fungsi tersebut tentang sumbu mana yang akan digunakan sebagai input. Karena

hasilnya dapat dipahami sebagai distribusi probabilitas, Softmax sering digunakan sebagai aktivasi untuk lapisan terakhir jaringan klasifikasi (Zhu et al., 2021).



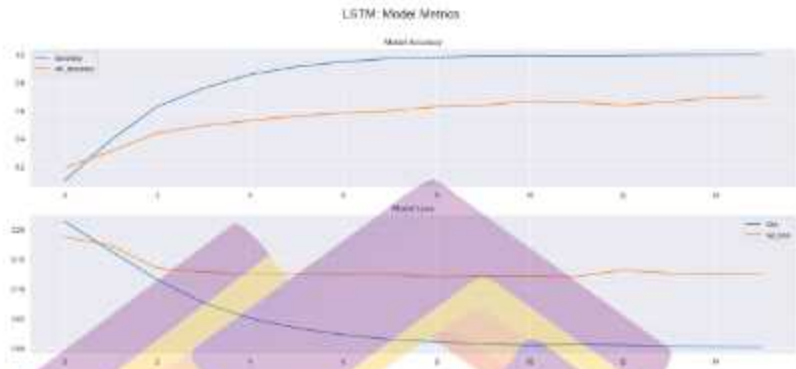
Gambar 9. Ringkasan Model Long-Short Term Memory

Hasil dari penelitian ini mendapatk 2 skenario. Yaitu skenario menggunakan data train dan skenario yang menggunakan data test. Pada tabel 2 dan grafiknya pada gambar 10 adalah hasil penelitian dari data train yang klasifikasikan dengan model LSTM mendapatkan hasil 16 epoch. Ini adalah hasil dari pemodelan LSTM. Yaitu hasil epoch training akurasi dan loss. Epoch adalah hiperparameter yang mendefinisikan berapa kali algoritme pembelajaran akan bekerja melalui seluruh dataset pelatihan. Satu epoch berarti bahwa setiap sampel dalam dataset pelatihan memiliki kesempatan untuk memperbarui parameter model internal (Hastomo et al., 2021).

Tabel 4. 1 Hasil Epoch Training Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.0966	0.1882	0.2140	0.1881
2	0.3886	0.3083	0.1641	0.1731
3	0.6291	0.4390	0.1155	0.1348
4	0.7608	0.4912	0.0763	0.1283
5	0.8572	0.5274	0.0502	0.1246
6	0.9134	0.5601	0.0338	0.1249
7	0.9487	0.5830	0.0225	0.1236
8	0.9712	0.5972	0.0150	0.1244
9	0.9802	0.6263	0.0109	0.1202
10	0.9891	0.6369	0.0071	0.1220
11	0.9924	0.6670	0.0049	0.1215
12	0.9884	0.6599	0.0062	0.1205
13	0.9920	0.6378	0.0052	0.1317
14	0.9960	0.6634	0.0031	0.1261
15	0.9966	0.6917	0.0020	0.1237
16	0.9977	0.6979	0.0015	0.1256

Berikut adalah hasil grafiknya.



Gambar 10. Model train LSTM

Selanjutnya adalah hasil penelitian yang menggunakan data train yang menggunkan fitur ekstraksi Glove telah di klasifikasikan dengan model LSTM. Menghasilkan hasil 12 epoch pada tabel 4.2 dan grafiknya di gambar 11.

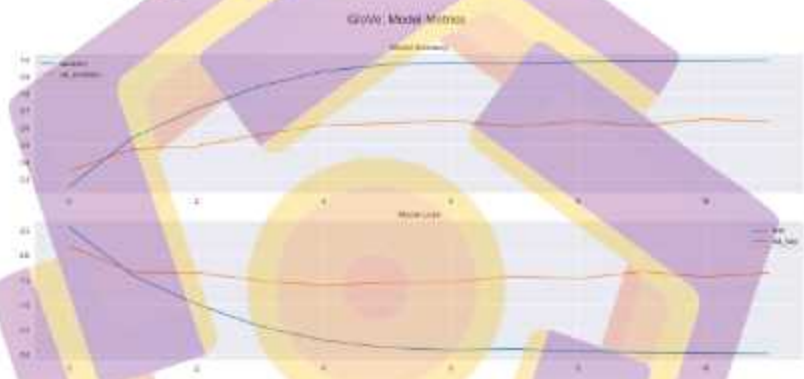
Tabel 4. 2 Hasil Fitur Ekstraksi Glove dengan Epoch Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.2561	0.3516	2.5616	2.1819
2	0.5444	0.4761	1.5934	1.6565
3	0.7105	0.4929	1.0154	1.6471
4	0.8444	0.5618	0.5699	1.4877
5	0.9314	0.6157	0.2884	1.3956
6	0.9707	0.6263	0.1416	1.4546
7	0.9809	0.6431	0.0926	1.4575
8	0.9744	0.6069	0.1140	1.5716

Tabel 4. 2 Lanjutan

9	0.9867	0.6387	0.0619	1.5251
10	0.9896	0.6140	0.0420	1.6699
11	0.9918	0.6502	0.0323	1.5637
12	0.9950	0.6387	0.0196	1.6444

Dengan grafik sebagai berikut.



Gambar 11. Hasil Train data dengan Glove

Hasil penelitian menggunakan data train yang menggunakan fitur ekstraksi Fasttext telah di klasifikasikan dengan model LSTM adalah sebagai berikut. Menghasilkan 11 epoch di table 4.3 dan grafik di gambar 12.

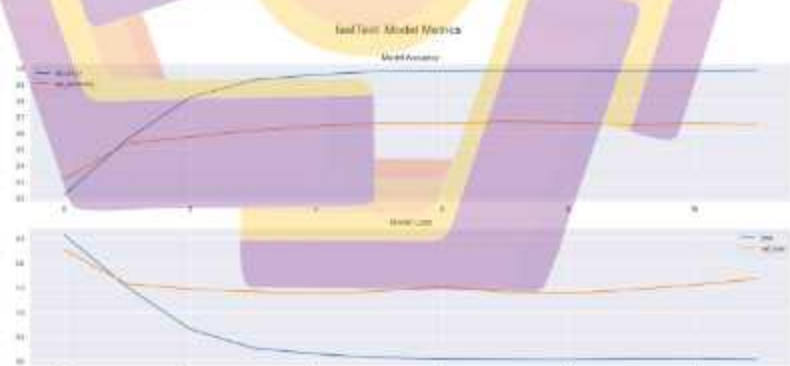
Tabel 4. 3 Hasil Fitur Ekstraksi FastText dengan Epoch Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.2594	0.3304	2.4652	2.1554

Tabel 4.3 Lanjutan

2	0.6030	0.5671	1.3419	1.3419
3	0.8310	0.6237	0.5988	1.2462
4	0.9287	0.6767	0.2655	1.1624
5	0.9676	0.6864	0.1247	1.2099
6	0.9770	0.6943	0.0916	1.2504
7	0.9852	0.7147	0.0563	1.1870
8	0.9864	0.6837	0.0473	1.4185
9	0.9822	0.6643	0.0627	1.4126
10	0.9917	0.7014	0.0331	1.3139
11	0.9940	0.7032	0.0207	1.3386

Berikut adalah grafiknya.



Gambar 12. Hasil Train data dengan Fasttext

Dari hasil penelitian yang menggunakan data train, dengan membandingkan epoch dari fitur ekstraksi Glove dan FastText dan diklasifikasikan dengan model LSTM. Untuk epoch pada fitur ekstraksi glove dengan 12 epoch, akurasinya berangsur stabil setelah epoch 4. GloVe menunjukkan bagaimana cara melibatkan informasi statistik global yang terkandung dalam dokumen. Dan dari beberapa referensi penelitian sebelumnya mengemukakan bahwa GloVe tidak mampu merepresentasikan vektor dari kata yang tidak ada dalam korpus (out of vocabulary) (Badri et al., 2022).

Selanjutnya adalah hasil penelitian yang menggunakan data train dengan fitur ekstraksi FastText menghasilkan 11 epoch. Akurasinya cenderung berangsur baik di epoch 3 dan seterusnya. FastText memiliki kemampuan memberikan representasi kata yang tidak muncul dalam data latih atau mampu mengatasi permasalahan out of vocabulary (Lim et al., 2019). Dari kedua percobaan pada data train kinerja fastText paling unggul untuk percobaan ini

Setelah menggunakan data train, selanjutnya melakukan percobaan menggunakan data pada subset test. Yang pertama adalah hasil dari model LSTM mendapatkan hasil 13 epoch.

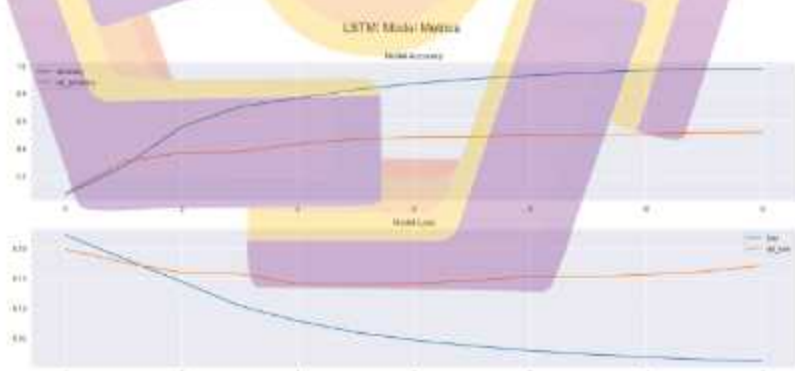
Tabel 4. 4 Hasil Epoch Testing Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.0649	0.0769	0.2221	0.1983
2	0.2703	0.3011	0.1825	0.1761
3	0.5580	0.3714	0.1438	0.1589

Tabel 4.4 Lanjutan

4	0.7023	0.3767	0.1032	0.1568
5	0.7663	0.4324	0.0782	0.1405
6	0.8266	0.4682	0.0591	0.1401
7	0.8725	0.4828	0.0461	0.1411
8	0.9054	0.4841	0.0364	0.1459
9	0.9305	0.4987	0.0284	0.1522
10	0.9519	0.5027	0.0224	0.1515
11	0.9680	0.5053	0.0175	0.1556
12	0.9768	0.5159	0.0134	0.1609
13	0.9804	0.5172	0.0111	0.1707

Berikut adalah grafiknya.



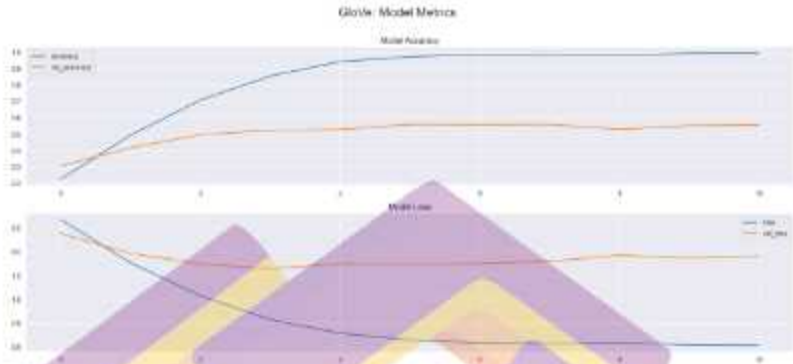
Gambar 13. Model Data Test LSTM

Hasil penelitian menggunakan data est yang menggunakan fitur ekstraksi Glove telah di klasifikasikan dengan model LSTM adalah sebagai berikut. Menghasilkan 11 epoch di tabel 4.5 dan grafik di gambar 14.

Tabel 4. 5 Hasil Fitur Ekstraksi Glove dengan Epoch Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.2251	0.3064	2.6765	2.3810
2	0.4928	0.4151	1.7690	1.9655
3	0.7066	0.4960	1.0777	1.7458
4	0.8536	0.5225	0.5748	1.6395
5	0.9413	0.5279	0.2873	1.7493
6	0.9715	0.5584	0.1505	1.7323
7	0.9854	0.5570	0.0858	1.7549
8	0.9872	0.5584	0.0672	1.7964
9	0.9811	0.5292	0.0886	1.9291
10	0.9922	0.5517	0.0423	1.8659
11	0.9926	0.5557	0.0345	1.9133

Berikut adalah grafiknya.



Gambar 14. Hasil Test data dengan fitur ekstraksi Glove

Selanjutnya adalah hasil penelitian yang menggunakan data test yang menggunkan fitur ekstraksi FastText telah di klasifikasikan dengan model LSTM. Menghasilkan hasil 12 epoch pada tabel 4.6 dan grafiknya di gambar 15.

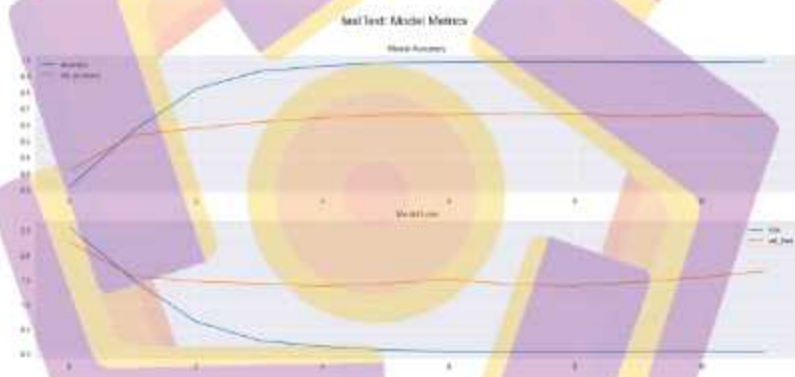
Tabel 4. 6 Hasil Fitur Ekstraksi FastText dengan Epoch Akurasi, Validasi Akurasi, Loss Dan Validasi Loss

Epoch	Akurasi	Validasi Akurasi	Loss	Validasi Loss
1	0.2191	0.3183	2.5718	2.2691
2	0.5617	0.5345	1.5095	1.5714
3	0.8199	0.5782	0.6557	1.4712
4	0.9280	0.6154	0.2730	1.4212
5	0.9605	0.6432	0.1515	1.3770
6	0.9813	0.6618	0.0767	1.4257
7	0.9857	0.6605	0.0521	1.5151

Tabel 4. 6 Lanjutan

8	0.9886	0.6711	0.0417	1.4065
9	0.9888	0.6658	0.0367	1.3839
10	0.9855	0.6512	0.0546	1.4587
11	0.9847	0.6631	0.0598	1.5498
12	0.9894	0.6538	0.0379	1.6794

Berikut adalah grafiknya.



Gambar 15. Hasil Test data dengan Fasttext

Dari hasil penelitian yang menggunakan data test, dengan membandingkan epoch dari fitur ekstraksi Glove dan FastText dan diklasifikasikan dengan model LSTM. Untuk epoch pada fitur ekstraksi glove dengan 11 epoch, akurasinya berangsur stabil setelah epoch 4 sama halnya sepertiyang dilakukan pada percobaan data train.

Selanjutnya adalah hasil penelitian yang menggunakan data train dengan fitur ekstraksi FastText menghasilkan 12 epoch. Akurasinya cenderung berangsur baik di epoch 3 dan seterusnya, percobaan ini pun bagus seperti yang menggunkan data train. Kinerja yang mempengaruhi algoritma LSTM yang baik dari eksperimen diperoleh dengan menggunakan word embedding FastText.

4.6 Evaluasi

Setelah melakukan beberapa percobaan seperti hasil yang telah dijelaskan sebelumnya, selanjutnya adalah evaluasi dari kedua fitur ekstraksi yang telah diklasifikasikan. Model evaluasi menggunakan format akurasi, presisi, recall dan f1-score untuk memberikan gambaran objektif tentang kinerja model. Evaluasi model membantu mengetahui seberapa baik model yang telah dibuat dalam memberikan hasil yang tepat (Yang et al., 2020). Paada gambar 16 menjelaskan model evaluasi yg digunakan dalam penelitian ini. Tabel 8 adalah rangkuman hasil daru percobaan menggunkan data test.

```
# Evaluate model on the test set
loss, accuracy, precision, recall = model.evaluate(X, y, verbose=0)
# Print metrics
print('')
print('Accuracy  : {:.4f}'.format(accuracy))
print('Precision : {:.4f}'.format(precision))
print('Recall    : {:.4f}'.format(recall))
print('F1 Score  : {:.4f}'.format(precision, recall))
```

Gambar 16. Model Evaluasi

Tabel 4. 7 Hasil percobaan menggunkan Data Train

Subset Data Dan Model	Akurasi	Presisi	Recall	F1-Score
Data Train + model LSTM	0.9682	0.9706	0.9672	0.9706
Data Train + GloVe + LSTm	0.9611	0.9694	0.9579	0.9694
Data Train + FastText + LSTM	0.9649	0.9741	0.9623	0.9741

Hasil evaluasi yang menggunakan data train yang diklasifikasikan dengan LSTM mendapatkan nilai akurasi sebesar : 0.9682, presisi : 0.9706, recall: 0.9672, dan F1 score : 0.9706.

Hasil penelitian untuk data train menggunakan fitur ekstraksi Glove yang udah di klasifikasikan menggunakan model LSTM sebagai berikut mendapatkan nilai akurasi sebesar : 0.9611, presisi : 0.9741, recall: 0.9623, dan F1 Score : 0.9741.

Selanjutnya hasil evaluasi untuk data train menggunakan fitur ekstraksi Fasttext yang sudah di klasifikasikan menggunakan model LSTM sebagai berikut mendapatkan nilai akurasi sebesar 0.9649, presisi : 0.9589, recall: 0.9353, dan F1 score : 0.9589.

Tabel 4. 8 Hasil percobaan menggunkan Data Test

Subset Data Dan Model	Akurasi	Presisi	Recall	F1-Score
Data Test + model LSTM	0.9417	0.9530	0.9385	0.9530
Data Test + GloVe + LSTm	0.9523	0.9651	0.9485	0.9651
Data Test + FastText + LSTM	0.9562	0.9701	0.9485	0.9701

Dari tabel 4.8 meringkas semuah hasil percobaan yang menggunakan data test. Setelah mendapatkan hasil penecobaan menggunakan data train, selanjutnya adalah evaluasi untuk data test. Hasil evaluasi yang menggunakan data train yang diklasifikasikan dengan LSTM mendapatkan nilai akurasi sebesar : 0.9417, presisi : 0.9530, recall: 0.9385, dan F1 score : 0.9530.

Hasil penelitian untuk data test menggunakan fitur ekstraksi Glove yang sudah di klasifikasikan menggunakan model LSTM sebagai berikut mendapatkan nilai akurasi sebesar : 0.9523, presisi : 0.9651, recall: 0.9519, dan F1 Score : 0.9651.

Sedangkan untuk model hasil untuk data test menggunakan fitur ekstraksi FastText yang sudah di klasifikasikan menggunakan model LSTM sebagai berikut mendapatkan nilai akurasi : 0.9562, presisi : 0.9701, recall : 0.9485 dan F1 Score : 0.9701.

LSTM dengan arsitektur GloVe dengan 200 dimensi menghasilkan 95,2%, untuk LSTM dengan arsitektur Fast Text dengan arsitektur 300 dimensi menghasilkan 95,6%.

Dari hasil penelitian ini membandingkan fitur ekstraksi GloVe dan FastText dapat disimpulkan bahwa fitur ekstraksi yg menggunkan Fasttext lebih baik dan efektif untuk melakukan klasifikasi teks. Perbedaan akurasi antara fastText dan GloVe word embedding sangat kecil. Kinerja terbaik dari eksperimen diperoleh dengan menggunakan word embedding FastText. Namun, perbedaan kinerja yang tidak begitu signifikan ini menunjukkan bahwa kedua word embedding ini memiliki kinerja yang kompetitif. Dan penggunaan dari fitur ekstraksi keduanya bergantung pada dataset yang digunakan dan permasalahan yg ingin diselesaikan.

Dalam penelitian yang dilakukan M. Alva Riza dkk. Dalam mendeteksi emosi pada sosial media twitter menggunakan LSTM dan Fasttext disana mereka membandingkan 3 fitur ekstraksi yaitu Word2vec, Glove, dan Fasttext dan mendapatkan hasil sebagi berikut Arsitektur word embedding, LSTM dengan Word2Vec dengan 50 unit dan 50 dropout menghasilkan 73,15%. Sebagai perbandingan, LSTM dengan arsitektur GloVe dengan 50 unit dan 30 dropout menghasilkan 60,1%, untuk LSTM dengan arsitektur Fast Text didapatkan arsitektur terbaik yaitu 50 unit dan 50 dropout menghasilkan 73,15%. LSTM memperoleh akurasi terbaik dengan Word2Vec dan LSTM dengan FastText, Fast Text memiliki kelebihan yaitu dapat menangani masalah out vocabulary, namun pada penelitian ini tidak dapat meningkatkan akurasi Word2Vec (Riza & Charibaldi, 2021).

Dalam penelitian lain yaitu mengklasifikasikan data berita palsu untuk Bahasa Indonesia dengan menggunakan fastText dan GloVe. Eksperimen ini memberikan kesimpulan sebagai berikut: 1) GRU memiliki kinerja yang lebih baik dibandingkan dengan LSTM ketika dataset memiliki frekuensi kemunculan yang lebih jarang dan tersebar luas; 2) Kedua model dua arah dari LSTM dan GRU menghasilkan nilai metrik yang lebih baik dibandingkan dengan model searah; 3) fastText lebih baik daripada GloVe dalam hal performa karena fastText dapat menangani kata-kata yang tidak ada di dalam kosa kata (OOV) dan kata-kata yang jarang ditemukan lebih baik daripada GloVe; 4) kombinasi fastText dan GRU dua arah memberikan hasil tertinggi dalam percobaan ini, terutama karena dataset tersebar luas dan memiliki panjang teks yang lebih pendek. Dalam penelitian ini, ditemukan masalah terkait dataset. Karena menggunakan bahasa sumber yang terbatas, berita satir lebih sulit ditemukan sehingga kami menggunakan berita dalam bahasa Inggris dan menerjemahkannya ke dalam bahasa Indonesia (Adipradana et al., 2021).

Pada penelitian perbandingan kinerja word embedding word2vec, glove, dan fasttext pada klasifikasi teks. Menurut temuan eksperimen, CNN memiliki kinerja 0.925, 0.958, dan 0.979 dalam hal klasifikasi teks dengan menggunakan penyematan kata Word2vec, GloVe, dan FastText dengan menggunakan F-Mean, masing-masing, untuk dataset 20 newsgroup dan 0.694, 0.688, dan 0.715 untuk dataset Reuters News. Vektor kata-kata yang tidak ada di dalam korpus (di luar kosakata) tidak dapat diwakili oleh Word2vec atau GloVe. Di sisi lain, FastText dapat diandalkan dalam kasus kekurangan kosakata ini. Dengan menggunakan

penyematan kata FastText, percobaan berjalan dengan baik. Ketiga penyematan kata tersebut berkinerja secara kompetitif, yang dibuktikan dengan perbedaan kinerja yang dapat diabaikan. Penerapannya sebagian besar bergantung pada masalah yang akan dipecahkan dan dataset yang digunakan (Nurdin et al., 2020).

Penelitian ini belum dapat menghasilkan akurasi yang sangat baik, hal ini dikarenakan data yang digunakan hanya menggunakan dari satu sumber dengan 20 class topik berbeda. Banyaknya class dalam dataset ini cukup berpengaruh terhadap hasil dari penelitian ini. Pada penelitian selanjutnya dapat membandingkan lebih banyak data sekaligus, untuk mendapatkan hasil yang lebih baik lagi, diharapkan dapat menerapkan metode deep learning yang lain, seperti CNN, BiLSTM, dll.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil percobaan yang telah dilakukan terhadap fitur ekstraksi Glove dan FastText yang diklasifikasikan dengan metode LSTM dapat disimpulkan bahwa:

a. Algoritma LSTM jika tidak dikombinasikan dengan fitur ekstraksi maka mendapatkan akurasi sebesar 94%, ini adalah akurasi yang tinggi. Banyaknya dataset dan waktu eksekusi dataset cukup berpengaruh dalam mengklasifikasikan teks terutama dalam hal ini algoritma LSTM.
b. Algoritma LSTM yang dikombinasikan dengan fitur ekstraksi Glove mencapai akurasi sebesar 95,2%. Hanya berpengaruh sedikit dari yang tidak dikombinasikan dengan fitur ekstraksi apapun.
c. Selanjutnya LSTM yang dikombinasikan dengan fitur ektraksi FastText mencapai akurasi sebesar 95,6%. Hanya memiliki sedikit perbedaan dengan fitur ekstraksi FastText. Perbedaan dari kedua fitur tidak begitu signifikan, yang menunjukkan kedua fitur ekstraksi memiliki kinerja yang sangat kompetitif.
d. Dari perbandingan kedua fitur ini yaitu Glove dan FastText, kinerja fasttext cukup berpengaruh terhadap algoritma LSTM. Namun fitur ekstraksi Glove juga memberikan performa terbaiknya.

5.2 Saran

Adapun beberapa saran yang direkomendasikan oeleh penulis untuk pengembangan penelitian tentang metode klasifikasi LSTM dalam permasalahan fitur ekstraksi adalah sebagai berikut:

1. Penelitian selanjutnya dapat mempertimbangkan untuk mengadopsi pendekatan yang lebih intensif dalam memaksimalkan penggunaan metode baik LSTM maupun metode lainnya untuk mencapai hasil yang optimal dengan model yang digunakan, terutama dalam konteks klasifikasi teks.
2. Penelitian selanjutnya dapat mempertimbangkan untuk pengujian metode klasifikasi menggunakan perbandingan dari beberapa dataset dan dataset berbahasa Indonesia.
3. Diharapkan penelitian selanjutnya dapat menampilkan pengujian menggunakan confusion matrix sehingga mendapatkan hasil yang lebih jelas dalam pengujian. Hal ini dapat memberikan wawasan yang lebih mendalam tentang hasil dari percobaan
4. Harapannya, penelitian selanjutnya menggunakan dimensi berbeda dari setiap fitur ekstraksi yang dapat menjadi pertimbangan untuk perbandingan.

DAFTAR PUSTAKA

PUSTAKA BUKU

Kulkarni, A., 2019, Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python, Apress, Bangalore, Karnataka, India

Beysolow, T., 2018, Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing, Apress, San Francisco, California, USA

Russel, S., 2010, Artificial Intelligent Techniques A Modern Approach, Fourth Edition, Pearson, USA

Anandarajan, M., 2019, Advances in Analytics and Data Science Practical Text Analytics Maximizing the Value of Text Data, Springer Nature, Switzerland

Bird, S., 2009, Natural Language Processing with Python, First Edition, O'Reilly Media, Gravenstein Highway North, Sebastopol

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Aa Zezen Zenal Abidin, E. Y. R. A. (2019). *PERINGKAS TEKS OTOMATIS DOKUEM TUNGGAL DAN MULTI BAHASA MENGGUNAKAN METODE TF-IDF*. 134–142.

Adipradana, R., Nayoga, B. P., Suryadi, R., & Suhartono, D. (2021). Hoax analyzer for indonesian news using rnns with fasttext and glove embeddings. *Bulletin of Electrical Engineering and Informatics*, *10*(4), 2130–2136. https://doi.org/10.11591/eei.v10i4.2956

Alshari, E. M., Azman, A., Doraisamy, S., Mustapha, N., & Alkeshr, M. (2017). Improvement of sentiment analysis based on clustering of Word2Vec features. *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, *2017-Augus*, 123–126. https://doi.org/10.1109/DEXA.2017.41

Badri, N., Kboubi, F., & Chaibi, A. H. (2022). Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection. *Procedia Computer Science*, *207*(Kes), 769–778. https://doi.org/10.1016/j.procs.2022.09.132

Beysolow II, T. (2018). Applied Natural Language Processing with Python Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing. In *Applied Natural Language Processing with Python*.

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

Buntoro, G. A. (2017). Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter. *Integer Journal*, *2*(1), 32–41. https://t.co/jrvaMsgBdH

Chen, C., & Dai, J. (2021). Mitigating backdoor attacks in LSTM-based text classification systems by Backdoor Keyword Identification. *Neurocomputing*, *452*, 253–262. https://doi.org/10.1016/j.neucom.2021.04.105

Chopra, S., Yadav, D., & Chopra, A. N. (2019). *Artificial Neural Networks Based Indian Stock Market Price Prediction: Before and After Demonetization*.

Curto, G., Jojoa Acosta, M. F., Comim, F., & Garcia-Zapirain, B. (2022). Are AI systems biased against the poor? A machine learning analysis using Word2Vec and GloVe embeddings. *AI and Society*, *Bradford 2020*. https://doi.org/10.1007/s00146-022-01494-z

D'Sa, A. G., Illina, I., & Fohr, D. (2020). BERT and fastText Embeddings for Automatic Detection of Toxic Speech. *Proceedings of 2020 International Multi-Conference on: Organization of Knowledge and Advanced Technologies, OCTA 2020*. https://doi.org/10.1109/OCTA49274.2020.9151853

Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. (2022). the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (Cnn) Text Classification. *Journal of Theoretical and Applied Information Technology*, *100*(2), 349–359.

Fudholi, D. H., Zahra, A., & Nayoan, R. A. N. (2022). A Study on Visual Understanding Image Captioning using Different Word Embeddings and CNN-Based Feature Extractions. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *4*(1), 91–98. https://doi.org/10.22219/kinetik.v7i1.1394

Gaikwad, V., & Haribhakta, Y. (2020). Adaptive glove and fasttext model for Hindi word embeddings. *ACM International Conference Proceeding Series*, 175–179. https://doi.org/10.1145/3371158.3371179

Gupta, P., Roy, I., Batra, G., & Dubey, A. K. (2021). *Decoding Emotions in Text Using GloVe Embeddings*. 36–40.

Hassani, A., Iranmanesh, A., & Mansouri, N. (2021). Text mining using nonnegative matrix factorization and latent semantic analysis. *Neural Computing and Applications*, *33*(20), 13745–13766. https://doi.org/10.1007/s00521-021-06014-6

Hastomo, W., Bayangkari Karno, A. S., Kalbuana, N., Meiriki, A., & Sutarno. (2021). Characteristic Parameters of Epoch Deep Learning to Predict Covid-19 Data in Indonesia. *Journal of Physics: Conference Series*, *1933*(1). https://doi.org/10.1088/1742-6596/1933/1/012050

Hochreiter, Sepp; Schmidhuber, J. (1997). Long Short-Term Memory. *Massachusetts Institute OfTechnology*, *50*(6), 2199–2207. https://doi.org/10.17582/journal.pjz/2018.50.6.2199.2207

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2019). Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Communications Magazine*, *57*(6), 114–119. https://doi.org/10.1109/MCOM.2019.1800155

Huang, Q., Zhang, F., & Li, X. (2018). *Machine Learning in Ultrasound Computer-Aided Diagnostic Systems : A Survey*. *2018*.

Ibrohim, M. O., & Budi, I. (2018). A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. *Procedia Computer Science*, *135*, 222–229. https://doi.org/10.1016/j.procs.2018.08.169

Indrapurasih, R. D., Bijaksana, M. A., Sardi, I. L., & Belakang, L. (2018). *Implementasi dan Analisis Kesamaan Semantik Antar Kata Bahasa Indonesia Menggunakan Metode GloVe Pendahuluan Studi Terkait Semantic Similarity*. *5*(3), 7699–7706.

Juwiantho, H., Setiawan, E. I., Santoso, J., Purnomo, M. H., Informasi, D. T., Tinggi, S., & Surabaya, T. (2020). Sentiment Analysis Twitter Bahasa Indonesia Berbasis WORD2VEC Menggunakan Deep Convolutional Neural Network. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *7*(1), 181–188. https://doi.org/10.25126/jtiik.202071758

K. joshi, A. (1991). *Natural Language Processing*. *253*.

Kulkarni, A., & Shivananda, A. (2019). Natural Language Processing Recipes. In *Natural Language Processing Recipes*. https://doi.org/10.1007/978-1-4842-4267-4

Kulkarni, A., & Shivananda, A. (2021). Natural Language Processing Recipes. In *Natural Language Processing Recipes*. https://doi.org/10.1007/978-1-4842-7351-7

Lim, E., Setiawan, E. I., & Santoso, J. (2019). Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning. *Journal of Intelligent System and Computation*, *1*(2), 65–73. https://doi.org/10.52985/insyst.v1i2.86

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*, 1–12.

Muhammad, P. F., Kusumaningrum, R., & Wibowo, A. (2021). Sentiment Analysis Using Word2vec and Long Short-Term Memory (LSTM) for Indonesian Hotel Reviews. *Procedia Computer Science*, *179*(2020), 728–735. https://doi.org/10.1016/j.procs.2021.01.061

Nurdin, A., Anggo Seno Aji, B., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*, *14*(2), 74. https://doi.org/10.33365/jtk.v14i2.732

Pennington, J., Socher, R., & Manning, C. D. (2014). *GloVe : Global Vectors for Word Representation*. 1532–1543.

Riza, M. A., & Charibaldi, N. (2021). Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text. *International Journal of Artificial Intelligence & Robotics (IJAIR)*, *3*(1), 15–26. https://doi.org/10.25139/ijair.v3i1.3827

Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020). Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi. *IOP Conference Series: Materials Science and Engineering*, *874*(1). https://doi.org/10.1088/1757-899X/874/1/012017

Sakketou, F., & Ampazis, N. (2020). A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons. *Knowledge-Based Systems*, *195*, 105628. https://doi.org/10.1016/j.knosys.2020.105628

Salur, M. U., & Aydin, I. (2020). A Novel Hybrid Deep Learning Model for Sentiment Classification. *IEEE Access*, *8*, 58080–58093. https://doi.org/10.1109/ACCESS.2020.2982538

Santos, I., Nedjah, N., & De Macedo Mourelle, L. (2017). Sentiment analysis using convolutional neural network with fasttext embeddings. *2017 IEEE Latin American Conference on Computational Intelligence, LA-CCI 2017 - Proceedings*, *2018-Janua*, 2–6. https://doi.org/10.1109/LA-CCI.2017.8285683

Shanita Biere Supervisor dr Sandjai Bhulai, A. (2018). *Hate Speech Detection Using Natural Language Processing Techniques*.

Sharma, Y., Agrawal, G., Jain, P., & Kumar, T. (2018). Vector representation of words for sentiment analysis using GloVe. *ICCT 2017 - International Conference on Intelligent Communication and Computational Techniques*, *2018-Janua*, 279–284. https://doi.org/10.1109/INTELCCT.2017.8324059

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Sklearn.datasets. (n.d.). *sklearn.datasets*. 2–3.

Sudiantoro, A. V., Zuliarso, E., Studi, P., Informatika, T., Informasi, F. T., Stikubank, U., & Mining, T. (2018). *Analisis Sentimen Twitter Menggunakan Text Mining Dengan Algoritma NAÏVE BAYES CLASSIFIER*. 398–401.

Susanty, M., & Sukardi, S. (2021). Perbandingan Pre-trained Word Embedding dan Embedding Layer untuk Named-Entity Recognition Bahasa Indonesia. *Petir*, *14*(2), 247–257. https://doi.org/10.33322/petir.v14i2.1164

Tandel, S. S., Jamadar, A., & Dudugu, S. (2019). A Survey on Text Mining Techniques. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, *Icaccs*, 1022–1026. https://doi.org/10.1109/ICACCS.2019.8728547

Trianto, R. B., Triyono, A., & Arum, D. M. P. (2020). Klasifikasi Rating Otomatis pada Dokumen Teks Ulasan Produk Elektronik Menggunakan Metode N-gram dan Naïve Bayes. *Jurnal Informatika Universitas Pamulang*, *5*(3), 295. https://doi.org/10.32493/informatika.v5i3.6110

Wan, C., Wang, Y., Liu, Y., Ji, J., & Feng, G. (2019). Composite Feature Extraction and Selection for Text Classification. *IEEE Access*, *7*(c), 35208–35219. https://doi.org/10.1109/ACCESS.2019.2904602

Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, 98–101. https://doi.org/10.1109/IWECAI50956.2020.00027

Yin, D., Lopes, R. G., Shlens, J., Cubuk, E. D., & Gilmer, J. (2019). *A Fourier Perspective on Model Robustness in Computer Vision*.

Zhu, C., Cai, S., Yang, Y., Xu, W., Shen, H., & Chu, H. (2021). A combined method for mems gyroscope error compensation using a long short-term memory network and kalman filter in random vibration environments. *Sensors (Switzerland)*, *21*(4), 1–21. https://doi.org/10.3390/s21041181