Laporan Projek Akhir

LSTM and GRU Networks for Ecommerce Product Text Classification

Disusun untuk memenuhi nilai Projek Akhir *Text Mining*

Disusun Oleh :

Kelompok 2

Edrick Setiawan	2540124021
Edward Federick	2540118624
Richard Gregorius	2501980961
Zaphenath Paneah Joseph Irawan	2501961520

Universitas Bina Nusantara

Kemanggisan

2024

BAB 1
PENDAHULUAN

1.1. Latar Belakang

Dalam era digital saat ini, *e-commerce* telah menjadi salah satu sektor yang berkembang pesat dan memiliki peran penting dalam perekonomian global. Pertumbuhan ini menyebabkan meningkatnya volume data produk yang tersedia secara online, baik dalam bentuk teks maupun deskripsi produk. Pengelolaan dan klasifikasi data teks ini menjadi tantangan tersendiri, terutama dalam memastikan bahwa produk dikategorikan dengan benar untuk memudahkan pencarian dan peningkatan pengalaman pengguna.

Metode tradisional untuk klasifikasi teks sering kali tidak mampu menangani kompleksitas dan variasi bahasa yang ada dalam deskripsi produk *e-commerce*. Metode seperti pengklasifikasian manual atau algoritma berbasis aturan memiliki keterbatasan dalam skala dan akurasi ketika berhadapan dengan volume data yang besar dan dinamis. Hal ini menyebabkan banyak perusahaan e-commerce mengalami kesulitan dalam menyediakan pengalaman pengguna yang optimal, di mana produk tidak selalu ditemukan di kategori yang tepat, sehingga mengurangi kepuasan pelanggan dan berpotensi menurunkan penjualan. Oleh karena itu, diperlukan pendekatan yang lebih canggih seperti penggunaan *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU).

1.2. Tujuan Penelitian

Penelitian ini bertujuan untuk mengembangkan model klasifikasi teks produk e-commerce yang lebih akurat dan efisien dengan menggunakan jaringan LSTM dan GRU. Kedua jenis jaringan ini dipilih karena kemampuannya dalam mengatasi masalah *long-term dependencies* yang sering muncul dalam data teks. Dengan memanfaatkan arsitektur LSTM dan GRU, penelitian ini diharapkan dapat meningkatkan akurasi klasifikasi teks produk *e-commerce* dibandingkan dengan metode konvensional.

1.3. Analisis Penelitian Terkait

Penelitian mengenai penggunaan *artificial neural networks* untuk klasifikasi teks telah banyak dilakukan dan dapat memberikan kinerja yang unggul dalam tugas-tugas klasifikasi teks. Salah satu penelitian terbaru yang relevan adalah studi yang dilakukan oleh Wang et al. (2023) dalam artikel "*An Improved LSTM-Based Failure Classification Model for Financial Companies Using Natural Language Processing*" yang diterbitkan di jurnal *Applied Sciences*. Penelitian tersebut membahas model LSTM yang dioptimalkan untuk klasifikasi kegagalan dalam perusahaan keuangan menggunakan pemrosesan bahasa alami (NLP). Hasil penelitian menunjukkan bahwa model LSTM dengan lapisan atensi mencapai akurasi lebih dari 92%. Studi ini memberikan bukti bahwa LSTM dan GRU dapat secara efektif diterapkan dalam tugas-tugas klasifikasi teks yang kompleks, termasuk dalam konteks *e-commerce*. Penelitian

ini tidak hanya menyoroti keunggulan LSTM dalam memahami konteks teks yang kompleks, tetapi juga menggarisbawahi pentingnya optimasi model untuk mencapai kinerja yang lebih baik.

Selain itu, penelitian yang dilakukan oleh Jansen (2020) dalam tesisnya di *Tilburg University* juga memberikan wawasan penting mengenai penggunaan LSTM dan GRU dalam klasifikasi teks. Jansen mengeksplorasi berbagai arsitektur *neural networks* untuk meningkatkan kinerja klasifikasi teks dalam berbagai domain, termasuk *e-commerce*. Temuan Jansen menunjukkan bahwa kombinasi LSTM dan GRU dapat menghasilkan peningkatan signifikan dalam akurasi dan efisiensi pemrosesan teks. Dalam penelitiannya, Jansen membandingkan berbagai model *neural networks* dan menemukan bahwa penggunaan LSTM dan GRU secara bersamaan dapat memanfaatkan keunggulan masing-masing arsitektur untuk mencapai hasil yang lebih optimal dalam klasifikasi teks.

1.4. Data Penelitian

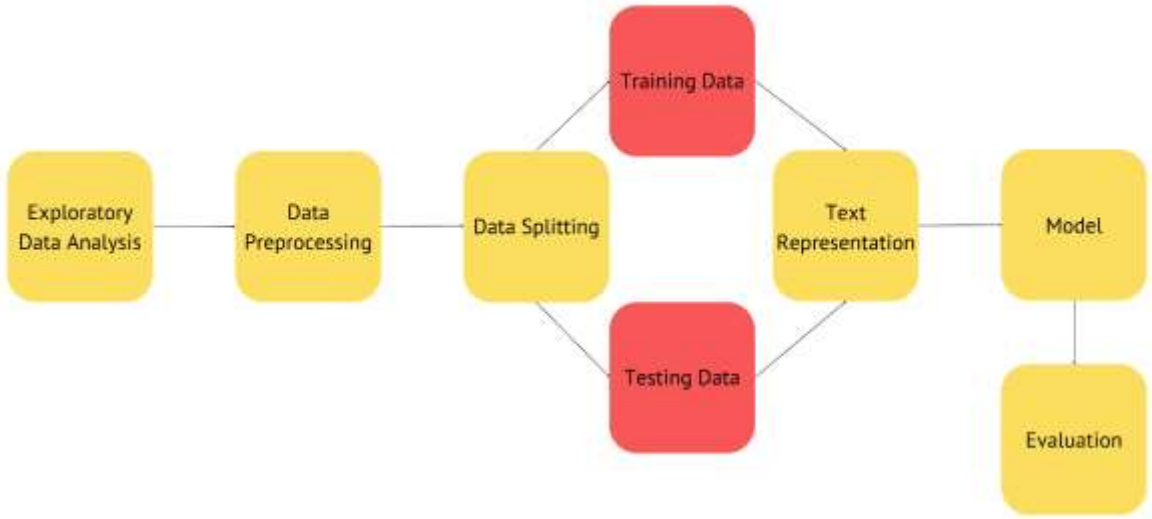
Data yang digunakan dalam penelitian ini adalah data teks terkait produk *e-commerce* yang bersumber dari Kaggle. Dataset ini dirancang untuk memfasilitasi klasifikasi teks-teks tersebut ke dalam berbagai kategori produk. Dataset ini terdiri dari 34680 baris dengan dua kolom utama yang mencakup label dan teks. Label dalam dataset ini mengategorikan produk ke dalam berbagai jenis, seperti "*Household*", "*Books*", "*Electronics*", dan "*Clothing & Accessories*". Jumlah produk dalam masing-masing kategori adalah sebagai berikut :

- "*Household*" memiliki 19313 baris
- "*Books*" memiliki 11820 baris
- "*Electronics*" memiliki 10621 baris
- "*Clothing & Accessories*" memiliki 8670 baris

BAB 2

METODOLOGI

2.1. Metode Penyelesaian Masalah



Tabel 2.1 Alur Penyelesaian Masalah

2.2. Proses Penyelesaian Masalah

a. Exploratory Data Analysis

Exploratory Data Analysis (EDA) adalah proses penting dalam analisis data yang bertujuan untuk memahami karakteristik utama dari data yang dimiliki sebelum melakukan analisis lebih lanjut. Proses ini melibatkan beberapa teknik seperti penanganan *missing value, resampling data*, visualisasi data, dan analisis statistik dasar.

Missing values dalam dataset dapat menyebabkan masalah dalam analisis dan pemodelan. Menghapus *missing values* adalah salah satu cara untuk menangani data yang tidak lengkap. Setelah melakukan penanganan pada *missing values*, langkah selanjutnya adalah melakukan *resampling* untuk mengubah jumlah sampel dalam dataset dikarenakan dataset yang kami gunakan cukup besar. Kami melakukan *resampling* dataset menjadi 50% dari ukuran aslinya. Selanjutnya, kami melakukan visualisasi data dengan countplot untuk memberikan gambaran jumlah sampel di setiap kategori. Selain countplot, kami juga melakukan analisis outlier dengan boxplot. Terakhir, kami juga menggunakan fungsi `*describe()*` untuk menghasilkan ringkasan statistik dasar dari dataset yang mencakup informasi seperti mean, median, *standard deviation*, dan *quartiles*.

b. Data Preprocessing

Data preprocessing merupakan langkah penting dalam pemrosesan data yang bertujuan untuk meningkatkan kualitas dan konsistensi data sebelum digunakan dalam model pembelajaran mesin. Tahap ini melibatkan berbagai teknik untuk membersihkan dan mengubah data mentah menjadi format yang lebih sesuai untuk analisis. Dalam konteks klasifikasi teks, data preprocessing mencakup beberapa langkah krusial seperti penghapusan spasi ganda, penghapusan data yang mengandung angka, penghapusan tanda baca, dan normalisasi teks menjadi lowercase.

Penghapusan spasi ganda dilakukan karena spasi ganda dalam teks dapat menyebabkan kesalahan dalam pemrosesan data, terutama ketika menghitung frekuensi kata atau melakukan tokenisasi. Oleh karena itu, langkah pertama dalam data cleansing adalah menghapus spasi ganda. Teknik ini dilakukan dengan menggantikan semua kemunculan spasi ganda dengan spasi tunggal dan kemudian memangkas spasi yang tidak diperlukan di awal dan akhir teks. Langkah selanjutnya yang dilakukan adalah menghapus data yang mengandung angka. Dalam banyak aplikasi klasifikasi teks, angka tidak memiliki arti penting dan dapat dihapus untuk menyederhanakan data. Penghapusan data yang mengandung angka dapat dilakukan dengan menggunakan ekspresi reguler yang mengidentifikasi dan menghapus angka dari teks.

Tanda baca sering kali tidak memberikan informasi yang relevan dalam analisis teks dan dapat menyebabkan *noise* dalam data. Penghapusan tanda baca bertujuan untuk membersihkan teks dari karakter-karakter ini. Hal ini bisa dilakukan dengan menggunakan pustaka string atau ekspresi reguler. Salah satu metode yang digunakan untuk menghapus tanda baca adalah dengan menggunakan ekspresi reguler `re.sub(r'[^\w\s]', '', text)`. Ekspresi reguler ini mencari semua karakter yang bukan huruf, angka, atau spasi, dan menggantinya dengan string kosong, sehingga menghapus semua tanda baca dari teks. Dan untuk mengurangi kompleksitas teks dan memastikan bahwa analisis tidak terpengaruh oleh *case sensitivity*, semua teks diubah menjadi huruf kecil (*lowercase*).

c. Data Splitting

Data splitting merupakan langkah penting dalam proses pengembangan model pembelajaran mesin. Tujuan utama dari *data splitting* adalah untuk memisahkan dataset menjadi beberapa subset yang berbeda untuk melatih dan menguji model. Ini memastikan bahwa model yang dikembangkan dapat generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya dan tidak hanya mengingat data pelatihan. Ada beberapa teknik umum untuk data splitting, salah satunya adalah *train-test split*.

Train-test split adalah metode dasar dalam *data splitting* di mana dataset dibagi menjadi dua bagian yaitu data pelatihan dan data pengujian. Data pelatihan digunakan untuk melatih model, sementara data pengujian digunakan untuk mengevaluasi kinerja model. Pembagian yang dilakukan dalam penelitian ini adalah 80% untuk pelatihan dan 20% untuk pengujian.

d. Text Representation

Text representation atau dalam hal ini vektorisasi teks adalah proses mengubah data teks menjadi representasi numerik yang dapat digunakan oleh model pembelajaran mesin. Dua metode populer untuk vektorisasi teks adalah TF-IDF (*Term Frequency-Inverse Document Frequency*) dan Word2Vec. Kedua metode ini memiliki pendekatan dan tujuan yang berbeda dalam representasi teks, namun sama-sama bertujuan untuk menangkap informasi penting dari data teks.

TF-IDF adalah metode statistik yang digunakan untuk mengevaluasi seberapa penting suatu kata dalam sebuah dokumen relatif terhadap koleksi dokumen (korpus). Metode ini menggabungkan dua ukuran yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF).

- ***Term Frequency*** (TF) mengukur frekuensi kemunculan kata dalam sebuah dokumen. TF dari suatu kata t dalam dokumen d dihitung dengan rumus :

$$TF(t,d) = \frac{Jumlah\ kemunculan\ t\ dalam\ d}{Jumlah\ kata\ dalam\ d}$$

- ***Inverse Document Frequency*** (IDF) mengukur seberapa umum atau jarang suatu kata dalam seluruh korpus. IDF dari suatu kata t dihitung dengan rumus :

$$IDF\ (t) = \log\left(\frac{Total\ jumlah\ dokumen}{Jumlah\ dokumen\ yang\ mengandung\ t}\right)$$

- TF-IDF adalah hasil perkalian dari TF dan IDF. Rumusnya adalah :

$$TF - IDF\ (t,d) = TF(t,d)x\ IDF(t)$$

Word2Vec adalah teknik embedding kata yang dikembangkan oleh Mikolov et al. pada tahun 2013. Teknik ini menggunakan *artificial neural network* untuk memetakan kata-kata dalam korpus besar ke dalam vektor-vektor dalam ruang berdimensi rendah. Ada dua arsitektur utama dalam Word2Vec yaitu *Continuous Bag of Words* (CBOW) dan *Skip-gram*. Arsitektur yang dipakai dalam penelitian ini adalah *Continuous Bag of Words* (CBOW) yang dapat memprediksi kata pusat (*target word*) dari konteks sekitarnya (*context words*). Model ini cenderung lebih cepat karena memprediksi kata target menggunakan rata-rata dari kata konteks.

e. Modelling and Experimentation

Penelitian ini menggunakan model *Long Short-Term Memory* (LSTM) dan *Gated Recurrent Unit* (GRU). Kedua jenis jaringan ini dirancang untuk menangani data urutan dan memiliki kemampuan untuk mengingat informasi jangka panjang.

LSTM adalah jenis *Recurrent Neural Network* (RNN) yang diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997. LSTM dirancang untuk mengatasi masalah *vanishing gradient* yang sering terjadi pada RNN tradisional. Struktur utama LSTM terdiri dari *memori cell* yang berfungsi untuk menyimpan informasi untuk waktu

yang panjang, *input gate* yang berfungsi untuk mengontrol seberapa banyak informasi dari input saat ini yang akan disimpan dalam *memori cell*, *output gate* yang berfungsi untuk mengontrol seberapa banyak informasi dari *memori cell* yang akan digunakan untuk output saat ini, dan *forget gate* yang berfungsi untuk mengontrol informasi mana yang harus dibuang dari *memori cell*.

GRU adalah varian dari LSTM yang diperkenalkan oleh Cho et al. pada tahun 2014. GRU menyederhanakan arsitektur LSTM dengan menggabungkan *input gate* dan *forget gate* menjadi satu *update gate* dan menggabungkan *memori cell* dengan *output gate*.

f. Evaluation Metrics

Evaluation metrics adalah alat yang digunakan untuk mengukur kinerja model pembelajaran mesin. Metode ini membantu dalam mengevaluasi seberapa baik model bekerja pada data pengujian dan membantu dalam membandingkan berbagai model. Evaluation metrics yang digunakan dalam penelitian ini adalah akurasi.

Akurasi adalah proporsi prediksi yang benar terhadap total prediksi yang dibuat. Metrik ini memberikan gambaran umum tentang kinerja model. Akurasi dapat dihitung dengan rumus sebagai berikut :

$$Akurasi = \frac{Prediksi\ Benar}{Total\ Prediksi}$$

BAB 3
HASIL DAN ANALISA

3.1 Model LSTM

Tabel 3.1 Tabel Hasil Permodelan LSTM

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1/10	0.6886	0.7391	0.2858	0.9219
2/10	0.2036	0.9527	0.2228	0.9419
3/10	0.1339	0.9713	0.2861	0.9348
4/10	0.0881	0.9823	0.2464	0.9383
5/10	0.1057	0.9794	0.2504	0.9461
Early Stopping				
Akurasi Akhir : 0.9419				

Pada awal pelatihan (Epoch 1), model memiliki nilai loss yang relatif tinggi (0.6886) dan akurasi (0.7391), yang menunjukkan bahwa model baru mulai belajar dari data dan masih melakukan banyak kesalahan dalam prediksi. Pada Epoch 2, terlihat ada penurunan yang signifikan dalam nilai loss menjadi 0.2036, sementara akurasi meningkat tajam menjadi 0.9527. Penurunan drastis dalam nilai loss menunjukkan bahwa model dengan cepat menyesuaikan bobotnya untuk meminimalkan kesalahan prediksi dan peningkatan akurasi yang signifikan menunjukkan bahwa model semakin mampu mengklasifikasikan teks dengan benar. Pada Epoch 3, nilai loss terus menurun menjadi 0.1339, dan akurasi meningkat menjadi 0.9713. Meskipun penurunan dalam nilai loss tidak secepat dari Epoch 1 ke Epoch 2, ini tetap menunjukkan peningkatan yang stabil dalam kinerja model. Pada Epoch 4, nilai loss semakin menurun menjadi 0.0881, dan akurasi mencapai 0.9823. Pada tahap ini, model mulai mencapai tingkat akurasi yang sangat tinggi. Dan akhirnya pada epoch 5, loss model secara signifikan menurun dan akurasi meningkat, mencapai loss 0.1057 dan akurasi 0.9794 pada Epoch 5. Ini menunjukkan bahwa model LSTM mampu mempelajari pola dari data dengan sangat baik.

Model LSTM yang dilatih menunjukkan kinerja yang sangat baik dengan akurasi yang tinggi dan kemampuan generalisasi yang kuat. Penurunan nilai loss dan peningkatan akurasi selama

epoch menunjukkan bahwa model berhasil belajar dan mengklasifikasikan teks dengan efisien. Validasi akurasi yang tinggi juga menegaskan bahwa model mampu bekerja dengan baik pada data yang tidak terlihat

3.2 Model GRU

Tabel 3.2 Tabel Hasil Permodelan GRU

Epoch	Loss	Accuracy	Val_loss	Val_accuracy
1/10	0.1609	0.9643	0.2401	0.9423
2/10	0.1024	0.9779	0.2184	0.9437
3/10	0.0781	0.9846	0.3098	0.9423
4/10	0.0570	0.9886	0.2925	0.9455
5/10	0.0566	0.9883	0.3109	0.9419
Early Stopping				
Akurasi Akhir : 0.9437				

Pada awal pelatihan (Epoch 1), model GRU menunjukkan nilai loss sebesar 0.1609 dan akurasi sebesar 0.9643. Ini menunjukkan bahwa model GRU mulai dengan performa yang cukup baik, mengingat bahwa akurasi sudah berada di atas 0.96. Pada Epoch 2, nilai loss menurun menjadi 0.1024, sementara akurasi meningkat menjadi 0.9779. Penurunan signifikan dalam nilai loss menunjukkan bahwa model dengan cepat menyesuaikan parameter untuk meningkatkan prediksinya. Peningkatan akurasi juga mengindikasikan bahwa model GRU mampu mempelajari pola dari data dengan lebih baik. Pada Epoch 3, nilai loss semakin menurun menjadi 0.0781, dan akurasi mencapai 0.9846. Pada tahap ini, model terus menunjukkan peningkatan dalam kinerja. Namun, pada epoch berikutnya, kita melihat bahwa nilai loss tidak lagi menurun secara signifikan, melainkan stabil, dengan nilai loss sebesar 0.0570 pada Epoch 4 dan 0.0566 pada Epoch 5. Akurasi juga mencapai puncaknya pada sekitar 0.9883 hingga 0.9886.

Model GRU yang dilatih menunjukkan kinerja yang sangat baik dengan akurasi yang tinggi dan kemampuan generalisasi yang kuat. Penurunan nilai loss dan peningkatan akurasi selama epoch menunjukkan bahwa model berhasil belajar dan mengklasifikasikan teks dengan efisien. Meskipun nilai loss validasi sedikit meningkat pada epoch terakhir, akurasi validasi yang tetap tinggi menunjukkan bahwa model mampu bekerja dengan baik pada data yang tidak terlihat.

3.3 Perbandingan Model

Tabel 3.3 Tabel Perbandingan Model LSTM dan GRU

Model	LSTM	**GRU**
Akurasi Akhir	0.9419	**0.9437**

Berdasarkan tabel diatas, model GRU mencapai akurasi akhir sebesar 0.9437, sedikit lebih tinggi dibandingkan model LSTM yang mencapai akurasi 0.9419. Perbedaan akurasi ini menunjukkan bahwa GRU memiliki keunggulan kecil untuk klasifikasi dataset ini. Selama proses pelatihan, baik LSTM maupun GRU menunjukkan kestabilan dalam peningkatan akurasi dan penurunan loss. Meskipun nilai loss validasi sedikit meningkat pada epoch terakhir untuk kedua model, akurasi validasi tetap tinggi, menunjukkan bahwa kedua model mampu mempertahankan performa yang konsisten sepanjang pelatihan.

BAB 4
KESIMPULAN

Penelitian ini berhasil mengembangkan model klasifikasi teks produk e-commerce yang akurat menggunakan jaringan Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU). Hasil menunjukkan bahwa model GRU sedikit lebih unggul dengan akurasi akhir sebesar 0.9437 dibandingkan 0.9419 untuk LSTM. Kedua model menunjukkan kemampuan generalisasi yang baik dan stabilitas performa yang tinggi selama pelatihan, menandakan bahwa keduanya efektif dalam mengklasifikasikan teks e-commerce.

Untuk penelitian lebih lanjut, disarankan untuk mengeksplorasi arsitektur jaringan yang lebih kompleks dan melakukan hyperparameter tuning lebih mendalam untuk mengoptimalkan performa model. Selain itu, penggunaan teknik preprocessing data yang lebih canggih serta pengujian model pada dataset yang lebih besar dan beragam dapat membantu meningkatkan kemampuan generalisasi model. Penerapan teknik regularisasi seperti dropout atau batch normalization juga dianjurkan untuk lebih mengurangi risiko overfitting.

REFERENSI

[1] Wang Z., Kim S., Joe I. (2023). An Improved LSTM-Based Failure Classification Model for Financial Companies Using Natural Languange Processing. Journal of Applied Sciences. Hanyang University.

[2] Jansen V. (2020). Classification of User Intention in e-commerce using Clickstream Data. Tillburg University. School of Humanities and Digital Sciences.

[3] Mikolov T., Chen K., Corrado G., Dean J. (2013). Efficient Estimation of Word Representations in Vector Space. Computation and Languange. Cornell University.

[4] Hochreiter S., Schmidhuber J. (1997). Long Short-Term Memory. Neural Computation Volume 9 Issue 8.

[5] Chung J., Gulcehre C., Cho K., Bengio Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Neural and Evolutionary Computing. Cornell University.