

Emotional Text Detection dengan LongShort Term Memory (LSTM)

¹Muhamad Dwirizqy Wimbassa, ²Taswiyah Marsyah Noor, ³Salma Yasara, ⁴Vannessa, ⁵Tubagus Muhammad Arsyah, ⁶Abdiansah
Fakultas ilmu Komputer, Universitas Sriwijaya^{1,2,3,4,5}

09021382126126@student.unsri.ac.id¹, 09021282126050@student.unsri.ac.id²,
09021282126072@student.unsri.ac.id³, 09021382126144@student.unsri.ac.id⁴,
09021382126164@student.unsri.ac.id⁵, abdiansah@unsri.ac.id⁶

Abstract

Emotional Text Detection is a technique in natural language processing that aims to identify the emotions contained in conversations or text messages. The LSTM (Long Short-Term Memory) method is one of the techniques used in natural language processing to model and predict sequential data. In this study, we propose the use of the LSTM method for emotion detection in conversation. The dataset used is a conversational dataset that contains positive, negative, and neutral emotions. We process datasets using data pre-processing techniques such as tokenization, data cleansing and one-hot encoding. Then, we train the LSTM model on the processed dataset and obtain evaluation results using accuracy metrics. The experimental results show that the LSTM model can be used to detect emotions in conversation with a good degree of accuracy. In addition, we also conducted an analysis on the prediction results of the model and showed that the LSTM model can correctly identify emotions. In conclusion, the LSTM method can be used to detect emotions in conversation with a good degree of accuracy. This method can be used to improve user experience in chat applications and increase the effectiveness of human and machine interactions.

Keyword: Dataset, Emotion, LSTM (Long Short-Term Memory), Natural Language.

Abstrak

*Emotional text detection adalah suatu teknik dalam pemrosesan bahasa alami yang bertujuan untuk mengidentifikasi emosi yang terkandung dalam percakapan atau pesan teks. Metode LSTM (Long Short-Term Memory) adalah salah satu teknik yang digunakan dalam pemrosesan bahasa alami untuk memodelkan dan memprediksi data sekuensial. Dalam penelitian ini, kami mengusulkan penggunaan metode LSTM untuk deteksi emosi dalam percakapan. Dataset yang digunakan adalah dataset percakapan yang mengandung emosi positif, negatif, dan netral. Kami memproses dataset menggunakan teknik pra-pemrosesan data seperti tokenisasi, pembersihan data, dan pengkodean *one-hot*. Kemudian, kami melatih model LSTM pada dataset yang telah diproses dan memperoleh hasil evaluasi menggunakan metrik akurasi. Hasil eksperimen menunjukkan bahwa model LSTM dapat digunakan untuk mendeteksi emosi dalam percakapan dengan tingkat akurasi yang baik. Selain itu, kami juga melakukan analisis pada hasil prediksi model dan menunjukkan bahwa model LSTM dapat mengidentifikasi emosi dengan tepat. Dalam kesimpulan, metode LSTM dapat digunakan untuk deteksi emosi dalam percakapan dengan tingkat akurasi yang baik. Metode ini dapat digunakan untuk meningkatkan pengalaman pengguna pada aplikasi chat dan meningkatkan efektivitas interaksi manusia dan mesin.*

Kata Kunci: Dataset, Emotion, LSTM (Long Short-Term Memory), Natural Language.

I. PENDAHULUAN

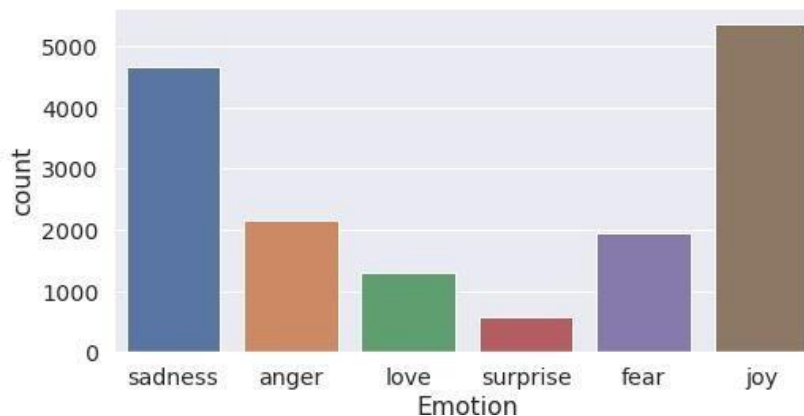
Emosi merupakan bagian dari perasaan yang berkaitan erat dengan kehidupan manusia. Emosi dapat diartikan sebagai perasaan atau keadaan mental yang melibatkan kegembiraan, kesedihan, rasa takut, kemarahan, atau perasaan lainnya yang dapat memengaruhi tindakan atau perilaku seseorang. Penggunaan emosi yang tepat pada waktu dan suasana yang tepat dapat mempengaruhi hasil kegiatan yang dilakukan oleh manusia[1]. Oleh karenanya, studi tentang pendeteksian emosi menjadi penting untuk dilakukan. Bentuk emosi verbal dapat dibagi menjadi dua, yaitu bentuk lisan ataupun tulisan. Saat ini, teks menjadi salah satu media utama dalam berkomunikasi. Secara garis besar, deteksi emosi manusia dalam teks diklasifikasikan dalam enam kelas di tingkat dasar, yaitu: *Love, Joy, Anger, Sadness, Fear* dan *Surprise*[2]. Mendeteksi emosi dalam teks dapat menjadi sumber informasi berharga, yang dapat digunakan untuk mempelajari bagaimana orang yang berbeda bereaksi terhadap situasi dan peristiwa yang berbeda[4]. Hal ini dapat diterapkan dalam berbagai bidang, seperti pengambilan keputusan dalam lingkungan sosial maupun lingkungan bisnis[4]. Namun, mendeteksi emosi dalam dialog tekstual adalah masalah yang menantang karena tidak adanya ekspresi wajah dan modulasi suara[5]. Terdapat juga

permasalahan lain, seperti teks yang sama dapat mengandung banyak emosi yang berbeda serta kata-kata yang digunakan dalam teks dapat memiliki makna ganda. Penelitian ini berfokus pada pendeteksian emosi dalam text. Dataset yang digunakan bersumber dari Kaggle, yakni Emotion Dataset for NLP[6], sedangkan pada metode, digunakan metode LSTM untuk mendeteksi emosi dalam teks dengan cara mengenali pola-pola tertentu yang terdapat dalam teks dan mengklasifikasikan teks tersebut ke dalam beberapa kategori emosi yang telah ditentukan.

II. METODE PENELITIAN

A. Dataset

Dataset yang digunakan untuk eksperimen pendeteksian emosi ini bersumber dari Kaggle, yaitu Emotion Dataset for NLP. Masing-masing data pada dataset ini telah dilabeli dengan berbagai macam jenis emosi, yakni *joy*, *sadness*, *anger*, *fear*, *love* dan *surprise*. Setiap label emosi memiliki jumlah data yang berbeda yaitu, *joy* 5362 data, *sadness* 4666 data, *anger* 2159 data, *fear* 1837 data, *love* 1304 data dan *surprise* 572 data. Untuk lebih jelas, jumlah data dalam setiap label emosi dapat dilihat pada grafik berikut:



Gambar 1. Jumlah Data per Emosi

Pada grafik dapat dilihat bahwa terdapat perbedaan jumlah data yang cukup signifikan antar jenis emosi, sehingga hasil deteksi dapat bervariasi dan tidak konsisten. Khususnya pada emosi *surprise* dan *love* dimana memiliki jumlah data terkecil. Pada dataset ini terdapat 20000 baris data yang dibagi menjadi 3 bagian yaitu, data test sebanyak 2000 baris, data train sebanyak 16000 baris, dan data validasi sebanyak 2000 baris.

B. Data preprocessing

Setelah mendapatkan dataset, data akan diproses untuk diambil informasi yang terkandung didalamnya[7]. Pada penelitian ini, preprocessing data dilakukan dengan cara pembersihan data (data cleaning) dan penyeragaman data (data transformation). Tujuan dari pembersihan data adalah agar data yang akan dimasukkan ke dalam model sesuai, meminimalkan noise dan dapat diolah pada tahap selanjutnya sehingga hasil klasifikasi yang diharapkan akan mendapatkan hasil yang maksimal, meminimalkan resiko eror[1]. Data yang dihapus berupa berbagai jenis komponen yang tidak dibutuhkan dalam pemrosesan data, salah satunya, yakni berupa stopwords. Stopwords adalah kata-kata yang disaring sebelum atau sesudah pemrosesan data bahasa alami(teks)[8]. Dilakukan juga text cleaning dan transformation text menjadi bentuk lower case.

C. LSTM(Long Short Term Memory)

Metode yang digunakan pada percobaan ini adalah metode LSTM(Long Short Term Memory). LSTM(Long Short Term Memory) merupakan salah satu jenis Recurrent Neural Network(RNN) yang merupakan modifikasi dari RNN. LSTM memiliki kemampuan untuk mengingat informasi dari masa lalu dalam waktu yang lama, sehingga dapat membantu dalam memprediksi hasil yang lebih baik pada data yang sangat kompleks. Dengan demikian, LSTM mampu mengingat kumpulan informasi yang telah disimpan dalam jangka waktu panjang, sehingga LSTM lebih efisien dalam memproses, memprediksi, sekaligus mengklasifikasikan data berdasarkan urutan waktu tertentu. LSTM diketahui lebih unggul dibanding metode neural network lainnya dikarenakan LSTM memiliki sistem gate tambahan, dengan menggunakan mekanisme gerbang (gate) untuk menangani masalah vanishing gradient pada Recurrent Neural Network dan juga untuk mengingat, membaca dan memperbarui informasi penting dalam jangka waktu yang panjang. Komponen pada LSTM antara lain adalah input gate, forget gate, cell gate, dan output gate[9].

D. Skenario Percobaan

Pada tahap ini, dilakukan 2 skenario percobaan, yakni skenario percobaan baseline dan modifikasi. Tahapan yang dilakukan, yakni setelah dikumpulkannya dataset yang akan digunakan, dataset tersebut lalu dibagi menjadi data train dan data test, selanjutnya, digunakan algoritma LSTM(Long Short-Term Memory) untuk melatih model menggunakan data train dan diuji menggunakan data test, selanjutnya performa model diukur menggunakan accuracy score. Pada tahap pembuatan baseline, model yang digunakan adalah model sequential dengan beberapa layer, layer yang digunakan antara lain, embedding, dropout, lstm dan dense yang dapat dilihat pada gambar berikut:

```

model = Sequential()
model.add(Embedding(input_dim=vocabSize, output_dim=150, input_length=300))
model.add(Dropout(0.2))
model.add(LSTM(128))
model.add(Dropout(0.2))
model.add(Dense(64, activation='sigmoid'))
model.add(Dropout(0.2))
model.add(Dense(6, activation='softmax'))
    
```

Gambar 2. pembuatan model percobaan baseline

Setelah model berhasil dibuat, terdapat summary berikut ini yang juga memuat total parameter dengan trainable params sejumlah 1, 951, 494 dan non-trainable params 0, yang berarti tidak ada satupun variabel yang tidak diperbarui atau dipelajari selama proses training.

Model: "sequential_17"

Layer (type)	Output Shape	Param #
embedding_16 (Embedding)	(None, 300, 150)	1800000
dropout_42 (Dropout)	(None, 300, 150)	0
lstm_23 (LSTM)	(None, 128)	142848
dropout_43 (Dropout)	(None, 128)	0
dense_29 (Dense)	(None, 64)	8256
dropout_44 (Dropout)	(None, 64)	0
dense_30 (Dense)	(None, 6)	390
Total params: 1,951,494		
Trainable params: 1,951,494		
Non-trainable params: 0		

Gambar 3. Summary model yang dibuat

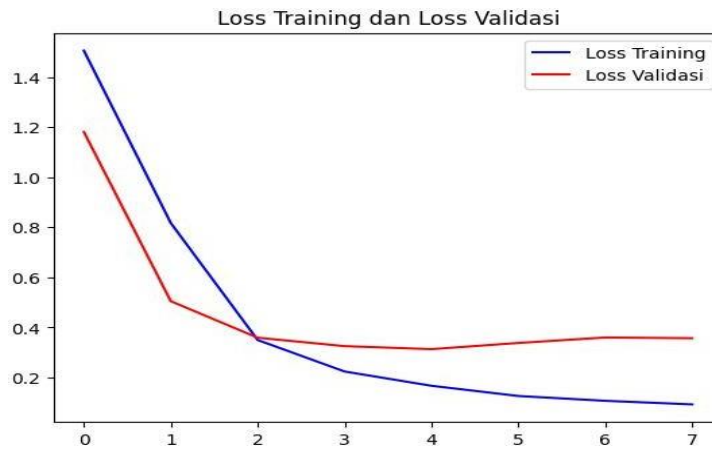
Setelahnya, dibuat callback yang bertujuan untuk mempersingkat waktu dalam membuat training. Setelah melakukan proses training, model yang diambil, ialah model epoch ke-5 yang memiliki val_loss terendah dengan tingkat akurasi training dari data train dan data validasi sebesar 87%.

III. HASIL PENELITIAN

Hasil penelitian menunjukkan bahwa penggunaan LSTM dalam pendeteksian emosi pada teks cukup berhasil. Persentase keberhasilan di ukur sebanyak 2 kali dengan menggunakan dua dataset yang berbeda. Pengukuran pertama dilakukan dengan dataset validasi dengan total 2000 line data, kemudian dilanjutkan dengan pengukuran dataset test yang juga memiliki total 2000 line data yang berbeda. Hasil dari pengukuran akurasi dan loss pada pengukuran ini dapat dilihat dari tabel dan gambar yang dicantumkan di bawah ini.



Gambar 4. Akurasi dataset validasi dan training



Gambar 5. Hasil evaluasi dataset validasi dan test

```

D >
#cek kualitas model dengan menggunakan data test dan val
lossVal, akurasiVal = modelKewren.evaluate(x_val, y_val)
lossTes, akurasiTes = modelKewren.evaluate(x_test, y_test)
[25]
... 63/63 [=====] - 4s 65ms/step - loss: 0.3250 - accuracy: 0.8845
63/63 [=====] - 4s 62ms/step - loss: 0.3611 - accuracy: 0.8750

D >
print("Loss dan Akurasi dari dataset Validasi : ", lossVal, akurasiVal)
print("Loss dan Akurasi dari dataset Test : ", lossTes, akurasiTes)

''' hasil akurasi dataset validasi sebesar 88,4%
    hasil akurasi dataset test sebesar 87,5% '''
[26]
... Loss dan Akurasi dari dataset Validasi : 0.32499009370803833 0.8845000267028809
Loss dan Akurasi dari dataset Test : 0.36110395193099976 0.875
    
```

Gambar 6. Hasil evaluasi dataset validasi dan test

Dari analisa Gambar 4, Gambar 5 dan Gambar 6. dapat dilihat bahwa akurasi dan loss dari dataset validasi dan test mendekati hasil dari training. Sumbu x merepresentasikan jumlah epoch dan sumbu y merepresentasikan persentase keberhasilan. Persentase keberhasilan pada dataset validasi mencapai 88% dan persentase keberhasilan pada dataset test menunjukkan angka 87,5%. Hal ini menunjukkan bahwa metode LSTM mampu mengenali emosi yang terkandung dalam teks dengan cukup baik.

Meskipun demikian, masih terdapat beberapa aspek yang perlu lebih dipelajari dan dioptimalkan. Salah satunya adalah penggunaan fitur-fitur yang lebih baik dalam menganalisis teks. Selain itu, pendekatan yang lebih holistik seperti menggunakan gambar atau suara sebagai sumber informasi emosi dapat menjadi fokus penelitian selanjutnya. Salah satu permasalahan ketika pembuatan model ini terdapat pada layer dropout.

Dropout layer adalah salah satu teknik regularisasi yang digunakan dalam model building untuk menghindari overfitting pada data training. Dalam dropout layer, sejumlah unit (atau node) acak pada layer sebelumnya akan di-"drop out" atau dinonaktifkan secara acak selama proses training. Hal ini mengurangi ketergantungan antar unit dan memperkecil risiko overfitting. Dropout layer telah terbukti efektif dalam memperbaiki akurasi model dan mengurangi overfitting pada dataset yang kompleks.

Pembuatan model ini menggunakan 3 dropout layer, yang tentunya berhasil dalam mengurangi kemungkinan overfitting data. Namun, efek samping dari penggunaan layer dropout adalah kurangnya konsistensi dalam pengklasifikasian emosi. Efek acak pada dropout layer yang banyak membuat model selalu berubah ketika model di muat. Pada proyek selanjutnya model dapat ditingkatkan dengan mengurangi penggunaan layer dropout

sehingga meningkatkan konsistensi dan memastikan model tidak memiliki perubahan ketika melakukan proses pemuatan.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa penggunaan LSTM dalam pendeteksian emosi pada teks memiliki potensi untuk dikembangkan lebih lanjut dan diaplikasikan pada berbagai aplikasi di masa depan.

IV. PEMBAHASAN

Model masih melakukan beberapa kesalahan dalam mendeteksi emosi dengan tingkat akurasi model 86%. Kesalahan dalam mendeteksi emosi terjadi karena terdapat perbedaan jumlah kata pada data. Contoh kesalahan deteksi dapat dilihat pada Gambar 7 dan Gambar 8.

```

deteksiEmosi("Im super afraid of spiders")
[9] ✓ 0.1s
... Im super afraid of spiders
['love'] : 0.8154386878013611
    
```

Gambar 7. Contoh 1 kesalahan deteksi

```

deteksiEmosi("Im surprised to see the principal at my house")
[10] ✓ 0.1s
... Im surprised to see the principal at my house
['anger'] : 0.8908125162124634
    
```

Gambar 8. Contoh 2 kesalahan deteksi

Untuk meminimalisir kesalahan model dilakukan modifikasi yang menghasilkan peningkatan akurasi sebesar 1% sehingga tingkat akurasi pun menjadi lebih tinggi yaitu sebesar 87%. Contoh dari hasil deteksi model setelah melalui tahap evaluasi dan modifikasi dapat dilihat pada Gambar 9 sampai Gambar 13.

```
▶ ▾ #silahkan mencoba!  
deteksiEmosi("I feel angry and furious")  
[68] ✓ 0.1s  
... I feel angry and furious  
['anger'] : 0.43175479769706726
```

Gambar 9. Contoh 1 setelah modifikasi model

```
▶ ▾ #silahkan mencoba!  
deteksiEmosi("my girlfriend is so sweet and caring, i dont know what i'll do without her")  
[51] ✓ 0.1s  
... my girlfriend is so sweet and caring, i dont know what i'll do without her  
['love'] : 0.2768385708332062
```

Gambar 10. Contoh 2 setelah modifikasi model

```
▶ ▾ #silahkan mencoba!  
deteksiEmosi("i feel so pissed today, i cant believe my ouw son lied to my face")  
[52] ✓ 0.1s  
... i feel so pissed today, i cant believe my ouw son lied to my face  
['surprise'] : 0.349923312664032
```

Gambar 11. Contoh 3 setelah modifikasi model

```
▶ ▾ #silahkan mencoba!  
deteksiEmosi("Im so excited for class today, its time to learn about AI")  
[54] ✓ 0.1s  
... Im so excited for class today, its time to learn about AI  
['joy'] : 0.5475074052810669
```

Gambar 12. Contoh 4 setelah modifikasi model

```
▶ ▾ #silahkan mencoba!  
deteksiEmosi("I keep feeling sad whenever i think about my cat that died, i miss her")  
[59] ✓ 0.1s  
... I keep feeling sad whenever i think about my cat that died, i miss her  
['sadness'] : 0.3455173671245575
```

Gambar 13. Contoh 5 setelah modifikasi model

V. KESIMPULAN

Dalam penelitian ini dilakukan modifikasi untuk meningkatkan tingkat akurasi dalam mendeteksi sebuah teks dengan cara melakukan pelatihan terhadap model, membagi dataset dan sebagainya agar sistem dapat mendeteksi berbagai jenis emosi pada teks. Berdasarkan hasil modifikasi yang telah dilakukan pada sistem tingkat akurasi meningkat mencapai 87% sehingga sistem telah dapat mendeteksi berbagai jenis emosi pada teks seperti, joy, sadness, fear, anger, love, dan surprise.

VI. REFERENSI

- [1] Akhmad Fadjeri, Hidayat, K., & Dwi Riska Handayani. (2021). EMOTION DETECTION ON TEXT USING NAIVE BAYES ALGORITHM. JURISTIK (Jurnal Riset Teknologi Informasi Dan Komputer), 1(02), 1-4. <https://doi.org/10.53863/juristik.v1i02.365>
- [2] W. Parrot, "Emotions in Social Psychology," in *Emotions in Social Psychology*, Psychology Press, 2001.
- [3] Rona Nisa Sofia Amriza, & Didi Supriyadi. (2021). Komparasi Metode Machine Learning dan Deep Learning untuk Deteksi Emosi pada Text di Sosial Media. JUPITER (Jurnal Penelitian Ilmu Dan Teknik Komputer), 13(2), 130–139. <https://doi.org/10.5281/3603.jupiter.2021.10>
- [4] Chatterjee, A., Narahari, K. N., Joshi, M., & Agrawal, P. (2019). SemEval-2019 Task 3: EmoContext Contextual Emotion Detection in Text. 39–48. <https://doi.org/10.18653/v1/s19-2005>
- [5] Fera Fanasya, Randy Cahya Wihandika, I. (2019). Deteksi Emosi pada Twitter Menggunakan Metode Naive Bayes dan Kombinasi Fitur. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(7), 3.
- [6] "Emotion dataset for NLP" <https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp?select=test.txt>
- [7] Harjanta, Aris Tri Jaka. (2015). Preprocessing Text untuk Meminimalisir Kata yang Tidak Berarti dalam Proses Text Mining. *Jurnal Informatika Upgris*, 1(6).
- [8] A. Rajaraman, J.D. Ullman, "Data Mining," *Mining of Massive Datasets*, pp. 1–17, 2011.
- [9] Purba, Meytry Petronella, & Yuliagnis Transver Wijaya. (2022). Analisis Basic Emotion Masyarakat Pada Masa Pandemi COVID-19 di Media Sosial Twitter Dengan Metode LSTM-FastText. *Seminar Nasional Official Statistics*, 1(1), 643-654. <https://doi.org/10.34123/semnasoffstat.v2022i1.1524>
- [10] "Emotion Classification(NLP)" <https://www.kaggle.com/code/dorgavra/emotion-classification-nlp>