



Laporan model klasifikasi teks menggunakan arsitektur RNN

Machine Learning 2

442023611050

Mohammad Ridho Cahyono

UNIVERSITAS DARUSSALAM GONTOR

PONOROGO

2025

1. Pendahuluan

Klasifikasi teks merupakan salah satu bidang penting dalam pemrosesan bahasa alami (NLP). Dalam era digital, data teks seperti ulasan aplikasi, komentar media sosial, atau email spam menjadi sumber informasi berharga. Tugas ini bertujuan membangun model klasifikasi berbasis RNN (LSTM) untuk membedakan ulasan aplikasi TikTok ke dalam beberapa kategori, sekaligus mengeksplorasi proses berpikir, eksperimen, dan refleksi pribadi dalam membangun model deep learning.

2. Dataset

Sumber Data:

Dataset dikumpulkan dari ulasan pengguna aplikasi TikTok di platform Google Play Store dan dikompilasi dalam file `Ulasan_Aplikasi_Tiktok.csv`.

Deskripsi Dataset:

- Total data: 5182
- Data elas:
 - 1451 ulasan bernada positif
 - 1651 ulasan bernada negatif
 - 2118 ulasan bernada netral
- Variasi: Terdapat perbedaan panjang teks, gaya bahasa (informal, bahasa campuran), hingga cara penyampaian opini.

Alasan Pemilihan:

Topik ini dipilih karena TikTok adalah aplikasi yang sangat populer dan memiliki banyak ulasan dari pengguna yang sangat beragam. Ini cocok untuk menguji kemampuan model dalam memahami konteks bahasa Indonesia dengan variasi ekspresi.

3. Implementasi Model

3.1 Arsitektur RNN

Model menggunakan arsitektur LSTM.

Model ini menggunakan layer embedding untuk representasi kata, LSTM untuk menangkap urutan kata, dan dense layer dengan aktivasi softmax untuk klasifikasi multi-kelas.

3.2 Preprocessing

- Tokenisasi menggunakan Tokenizer(num_words=5000)
- Padding menggunakan pad_sequences dengan panjang maksimum 200
- Label dikonversi ke one-hot encoding menggunakan to_categorical
- Data dibagi ke dalam training dan testing dengan rasio 80:20

3.3 Pengaturan Eksperimen

- Optimizer: Adam
- Loss: categorical_crossentropy
- Epoch: 10
- Batch size: default
- Validation data: 20%

3.4 Log Eksperimen

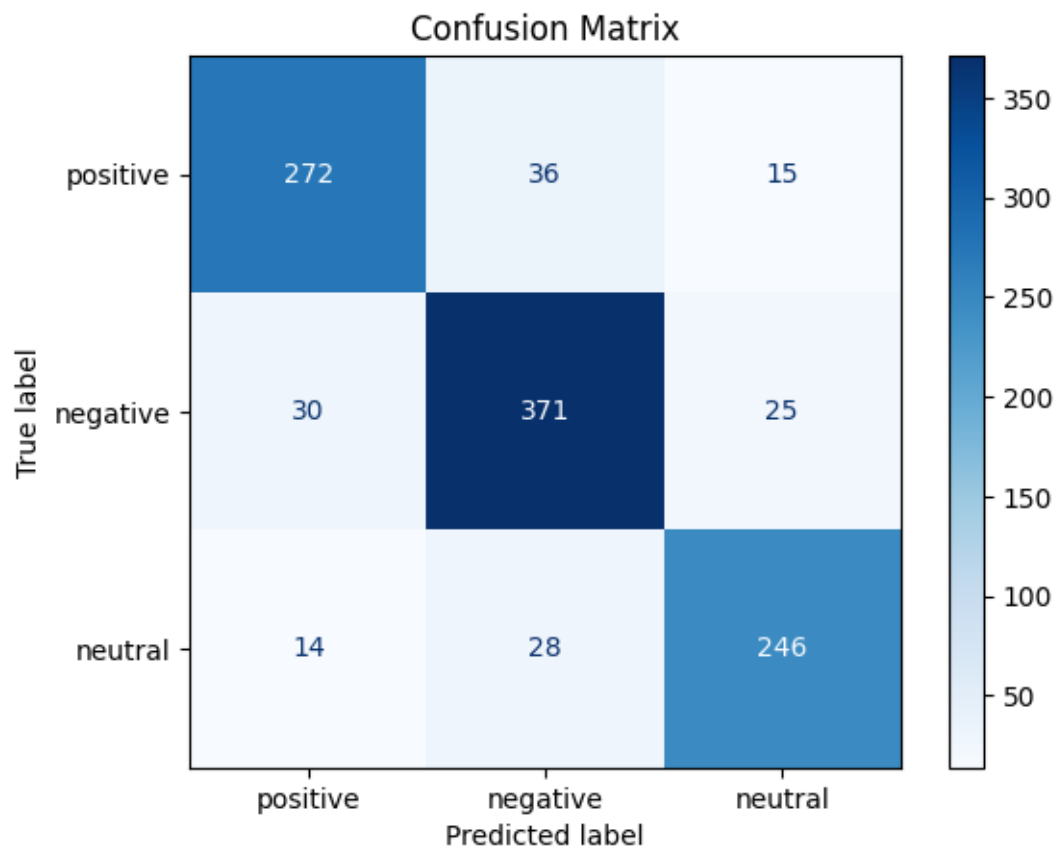
Percobaan Model Dropout Optimizer Akurasi Validasi Catatan

#1	LSTM(64) 0	Adam	86.21%	Overfitting dari epoch 4
#2	LSTM(64) 0.2	Adam	85.6%	Lebih stabil, hasil terbaik

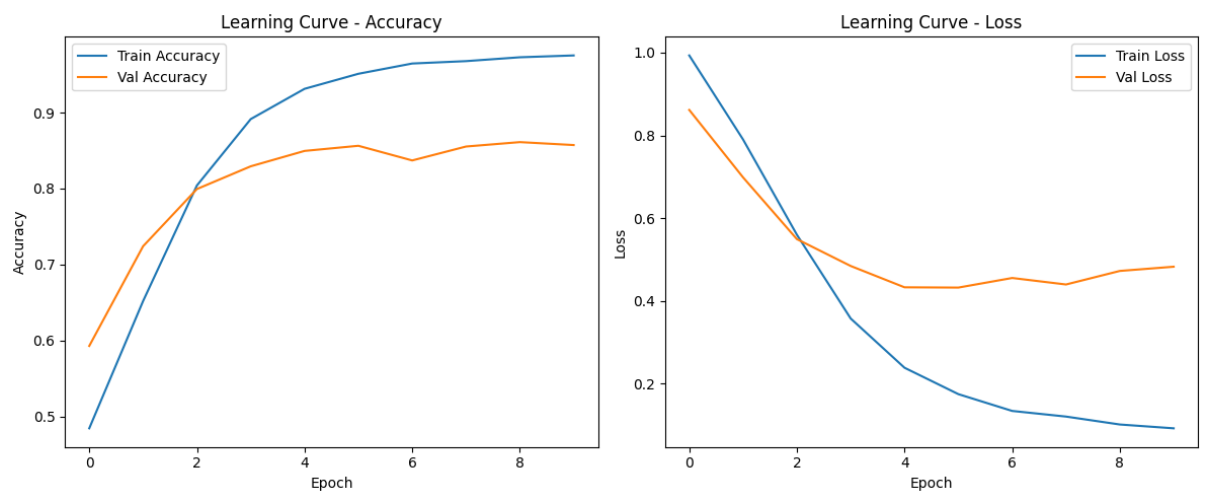
4. Evaluasi Hasil

- Akurasi Testing: 85.7%

- Confusion Matrix:



- Learning Curve:



Analisis:

Model kedua menunjukkan performa lebih baik karena dropout mencegah overfitting walau belum sempurna dikarenakan data yang tidak seimbang

5. Refleksi Pribadi

Tantangan utama adalah preprocessing, karena teks ulasan informal sulit untuk dinormalisasi. Saya mencoba beberapa metode dan akhirnya menggunakan pendekatan sederhana: lowercasing dan tokenisasi standar.

Pelajaran terpenting dari tugas ini adalah memahami proses end-to-end NLP mulai dari preprocessing hingga evaluasi model.

6. Kesimpulan dan Saran

Model LSTM dengan dropout 0.2 berhasil mencapai akurasi validasi 85%. Untuk pengembangan lebih lanjut:

- Gunakan pretrained embedding
 - Coba arsitektur BiLSTM atau GRU
 - Menyetarakan jumlah data per kelas
-

7. Referensi

- Keras Documentation: <https://keras.io>
- Towards Data Science Articles on LSTM
- Dataset pribadi hasil scraping ulasan TikTok