Laporan Tugas Individu: Klasifikasi Teks Menggunakan RNN (LSTM)

Name: Ahmad Mukhlis Farhan

NIM: 442023611001

1. Pendahuluan

Klasifikasi teks adalah salah satu tugas mendasar dalam bidang pemrosesan bahasa alami (NLP) yang memiliki banyak aplikasi nyata seperti deteksi spam, analisis sentimen, klasifikasi berita, dan lainnya. Dalam tugas ini, saya membangun model klasifikasi teks menggunakan arsitektur RNN, khususnya LSTM (Long Short-Term Memory), untuk memisahkan dua kelas teks: **spam** dan **non-spam**.

Tujuan dari tugas ini bukan hanya mendapatkan model dengan akurasi tinggi, tetapi juga memahami proses berpikir, eksplorasi parameter, serta refleksi terhadap hasil dan tantangan yang dihadapi.

2. Dataset

2.1 Sumber Data

Dataset diambil dari dataset SMS Spam Collection (sumber terbuka) yang berisi ribuan pesan teks yang diklasifikasikan sebagai "spam" dan "ham" (non-spam).

2.2 Deskripsi Dataset

- Jumlah data total: 2000
- Kelas:
    - Spam: 1000 data
    - Non-spam: 1000 data
- Variasi panjang teks: mulai dari 10 hingga 200 kata.
- Gaya bahasa: formal, informal, serta penggunaan emoji dan singkatan umum.

2.3 Alasan Pemilihan Dataset

- Klasifikasi spam adalah kasus nyata dengan relevansi tinggi.
- Dataset tersedia secara terbuka dan telah digunakan dalam banyak penelitian.
- Data mencerminkan tantangan dunia nyata: bahasa yang tidak baku, singkatan, dan pesan pendek.

3. Implementasi Model

3.1 Arsitektur RNN

- Model: LSTM (Long Short-Term Memory)
- Layers:

- Embedding layer
- LSTM layer (64 units, dropout=0.5)
- Dense layer (sigmoid activation)

3.2 Preprocessing

- Tokenisasi menggunakan `Tokenizer` dari Keras
- Padding dengan `pad_sequences`
- Label binarisasi (spam=1, non-spam=0)
- Split data: 80% training, 20% validation

3.3 Pengaturan Eksperimen

- Optimizer: Adam
- Loss Function: Binary Crossentropy
- Epoch: 10
- Batch size: 32

3.4 Log Eksperimen

Percobaan	Model	Dropout	Optimizer	Akurasi Validasi	Catatan
#1	LSTM(64)	0	Adam	96.4%	Overfitting sejak epoch 2
#2	LSTM(64)	0.5	Adam	98.3%	Lebih stabil, val loss menurun

---

4. Evaluasi Hasil

4.1 Learning Curve

- Akurasi training naik tajam hingga >99%.
- Akurasi validasi stabil sekitar 98–99%, namun loss validasi naik setelah epoch ke-3, menandakan overfitting ringan.

4.2 Confusion Matrix

	Pred: Non-Spam	Pred: Spam
Actual: Non-Spam	955	11
Actual: Spam	9	140

- **Precision**, **Recall**, dan **F1-score** tinggi menunjukkan performa baik.

5. Refleksi Pribadi

Tantangan:

- Menghindari overfitting meskipun akurasi validasi cukup tinggi.
- Menyusun pipeline preprocessing yang konsisten.

Solusi:

- Menambahkan dropout dan mencoba berbagai nilai `maxlen`.
- Stratified split untuk mempertahankan distribusi kelas.

Pembelajaran:

- Visualisasi learning curve membantu mendeteksi overfitting secara dini.
- Iterasi model dan eksperimen parameter penting untuk mendapatkan performa optimal.

Saya menggunakan ChatGPT untuk memverifikasi struktur laporan dan menyusun penjelasan reflektif secara akademis, namun semua kode, eksperimen, dan interpretasi dilakukan sendiri.

6. Kesimpulan dan Saran

Kesimpulan:

- Model LSTM berhasil mengklasifikasikan teks dengan **akurasi validasi 98.3%**.
- Walau akurasi tinggi, terjadi overfitting ringan yang terdeteksi dari val_loss.

Saran:

1. Implementasi **early stopping** untuk mencegah over-training.
2. Coba model **BiLSTM** atau **GRU** untuk variasi arsitektur.
3. Gunakan **pre-trained embedding** seperti GloVe untuk meningkatkan hasil.
4. Evaluasi lanjutan dengan precision, recall, F1.
5. Tambah variasi data atau gunakan augmentasi untuk memperkuat generalisasi.

7. Referensi

- Almeida, T.A., Hidalgo, J.M.G., & Yamakami, A. (2011). SMS Spam Collection Dataset.
- Chollet, F. (2018). Deep Learning with Python. Manning Publications.
- https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
- https://www.tensorflow.org/api_docs/python/tf/keras