

PEMROSESAN BAHASA ALAMI

Nama Anggota Kelompok :

1. Andini Nur Izzati (121450147)
2. Lulu Christin Sihombing (121450152)
3. Ditta Winanda Putri (121450006)
4. Jasmine Georgia Sekartaji (121450159)
5. Rizka Yustiana Zahra (121450058)

Analisis Perbandingan model di atas dengan parameter :

- Dataset (Apakah membutuhkan yang lebih besar?)
- Waktu dan Sumber Daya Komputasi
- Jelaskan Generalisasi

1. LSTM

Kode: https://colab.research.google.com/drive/1sHs3qXBJT_7B2GD0Ft_vdb0gjMgd1YW2

```
Epoch 1/25, Loss: 0.7886, Test Accuracy: 0.5050
Epoch 2/25, Loss: 0.7095, Test Accuracy: 0.4900
Epoch 3/25, Loss: 0.6881, Test Accuracy: 0.5000
Epoch 4/25, Loss: 0.6881, Test Accuracy: 0.4900
Epoch 5/25, Loss: 0.6780, Test Accuracy: 0.4650
Epoch 6/25, Loss: 0.6633, Test Accuracy: 0.5000
Epoch 7/25, Loss: 0.6678, Test Accuracy: 0.4650
Epoch 8/25, Loss: 0.6528, Test Accuracy: 0.4500
Epoch 9/25, Loss: 0.6413, Test Accuracy: 0.4600
Epoch 10/25, Loss: 0.6312, Test Accuracy: 0.4900
Epoch 11/25, Loss: 0.6022, Test Accuracy: 0.4750
Epoch 12/25, Loss: 0.5963, Test Accuracy: 0.5050
Epoch 13/25, Loss: 0.5815, Test Accuracy: 0.4750
Epoch 14/25, Loss: 0.5672, Test Accuracy: 0.5050
Epoch 15/25, Loss: 0.5454, Test Accuracy: 0.5300
Epoch 16/25, Loss: 0.5254, Test Accuracy: 0.5000
Epoch 17/25, Loss: 0.5122, Test Accuracy: 0.5350
Epoch 18/25, Loss: 0.5255, Test Accuracy: 0.4550
Epoch 19/25, Loss: 0.4849, Test Accuracy: 0.5250
Epoch 20/25, Loss: 0.4584, Test Accuracy: 0.4900
Epoch 21/25, Loss: 0.4362, Test Accuracy: 0.5000
Epoch 22/25, Loss: 0.4500, Test Accuracy: 0.5000
Epoch 23/25, Loss: 0.4285, Test Accuracy: 0.5150
Epoch 24/25, Loss: 0.3635, Test Accuracy: 0.5050
Epoch 25/25, Loss: 0.3580, Test Accuracy: 0.5150
```

- Dataset (Apakah membutuhkan yang lebih besar?)

Ya, penggunaan dataset yang lebih besar sering kali penting untuk meningkatkan performa model NLP, terutama saat menggunakan arsitektur seperti LSTM. Alasannya yaitu dataset besar membantu model menangkap lebih banyak variasi dalam data dan mengurangi overfitting. Embedding seperti Glove memanfaatkan co-occurrence statistik kata dalam korpus besar, tetapi jika dataset pelatihan terlalu

kecil, pemanfaatannya bisa kurang optimal. Model dengan banyak parameter (seperti LSTM) memerlukan banyak data agar bobot dapat dioptimalkan dengan baik.

Jika dataset saat ini memberikan akurasi akhir rendah (seperti yang terlihat dari hasil dengan akurasi 51.50%), mungkin dataset yang lebih besar diperlukan.

- **Waktu dan Sumber Daya Komputasi**

Model LSTM memerlukan lebih banyak waktu untuk pelatihan dibandingkan model sederhana seperti logistic regression. Tambahan embedding Glove juga menambah waktu inisialisasi. Eksperimen ini menggunakan PyTorch dan TensorFlow, yang dapat memanfaatkan GPU. Jika dataset lebih besar atau arsitektur model lebih kompleks (misalnya menambah lapisan LSTM), GPU menjadi krusial untuk mempercepat pelatihan.

- **Jelaskan Generalisasi**

Generalisasi merujuk pada kemampuan model untuk bekerja baik pada data yang belum pernah dilihat (data uji). Beberapa faktor yang mempengaruhi generalisasi: Dataset yang kecil bisa menyebabkan overfitting, di mana model belajar terlalu spesifik pada data pelatihan. Dropout (seperti yang diterapkan dengan 50% dalam eksperimen ini) membantu meningkatkan generalisasi dengan mencegah overfitting. Skor akurasi pada data uji adalah indikator umum generalisasi. Dalam eksperimen ini, skor akurasi yang rendah menunjukkan kemungkinan model kurang menangkap pola secara umum, mungkin karena dataset kecil atau masalah lainnya.

2. GRU

```
Epoch 1/10, Loss: 0.9163, Test Accuracy: 0.7698
Epoch 2/10, Loss: 0.3210, Test Accuracy: 0.8184
Epoch 3/10, Loss: 0.2803, Test Accuracy: 0.8037
Epoch 4/10, Loss: 0.2560, Test Accuracy: 0.7521
Epoch 5/10, Loss: 0.2368, Test Accuracy: 0.7744
Epoch 6/10, Loss: 0.2199, Test Accuracy: 0.7509
Epoch 7/10, Loss: 0.2054, Test Accuracy: 0.7310
Epoch 8/10, Loss: 0.1916, Test Accuracy: 0.7284
Epoch 9/10, Loss: 0.1787, Test Accuracy: 0.7222
Epoch 10/10, Loss: 0.1657, Test Accuracy: 0.7219
```

- **Dataset (Apakah membutuhkan yang lebih besar?)**

Dataset yang digunakan cukup untuk menghasilkan model dasar dengan akurasi 72.19%, tetapi performa yang optimal membutuhkan dataset yang lebih besar dan lebih beragam. Dataset yang kecil atau kurang representatif dapat menyebabkan model kesulitan mempelajari pola yang kompleks, yang terlihat dari tanda-tanda overfitting pada hasil pelatihan. Penambahan data latih, baik melalui pengumpulan data tambahan atau augmentasi teks, dapat membantu meningkatkan kemampuan generalisasi model dan meminimalkan overfitting.

- **Waktu dan Sumber Daya Komputasi**

Pelatihan model memakan waktu 37 menit pada GPU T4 untuk 10 epoch, menunjukkan efisiensi komputasi GRU dibandingkan model yang lebih kompleks. Namun, dengan akurasi pengujian yang stagnan, diperlukan optimasi tambahan, seperti tuning hyperparameter dan regularisasi, untuk meningkatkan performa tanpa memperpanjang waktu pelatihan secara signifikan.

- **Jelaskan Generalisasi**

Penurunan akurasi pengujian dari 0.8184 menjadi 0.7219 menunjukkan masalah generalisasi. Langkah seperti validasi silang, penggunaan dropout, dan regularisasi L2 dapat membantu model belajar pola yang lebih representatif, sekaligus mengurangi risiko penurunan performa pada data pengujian.

3. FastText

Epoch 1/10	Train Loss: 0.786	Test Loss: 0.778	Test Accuracy: 71.707%
Epoch 2/10	Train Loss: 0.745	Test Loss: 0.751	Test Accuracy: 72.276%
Epoch 3/10	Train Loss: 0.723	Test Loss: 0.734	Test Accuracy: 72.673%
Epoch 4/10	Train Loss: 0.710	Test Loss: 0.724	Test Accuracy: 72.858%
Epoch 5/10	Train Loss: 0.700	Test Loss: 0.717	Test Accuracy: 73.150%
Epoch 6/10	Train Loss: 0.695	Test Loss: 0.713	Test Accuracy: 73.189%
Epoch 7/10	Train Loss: 0.689	Test Loss: 0.707	Test Accuracy: 73.137%
Epoch 8/10	Train Loss: 0.685	Test Loss: 0.703	Test Accuracy: 73.520%
Epoch 9/10	Train Loss: 0.683	Test Loss: 0.700	Test Accuracy: 73.507%
Epoch 10/10	Train Loss: 0.680	Test Loss: 0.698	Test Accuracy: 73.613%

Kode :  TUGAS_TERAKHIR_NLP_FASTTEXT.ipynb

- **Dataset (Apakah membutuhkan yang lebih besar?)**

Dataset yang digunakan adalah `ag_news_csv` dengan total baris 119211 pada data pelatihan dan 7553 baris pada data uji. Dataset ini cukup besar untuk model seperti FastText, yang relatif ringan dibandingkan model berbasis Transformer. Namun, ukuran ini sudah cukup memadai untuk melatih model FastText secara efektif. Sehingga dengan jumlah dataset seperti yang telah disebutkan itu sudah cukup untuk membangun model dan menambah dataset tidak terlalu diperlukan. Dengan melakukan preprocessing pada data akan membantu dalam meningkatkan akurasi.

- **Waktu dan Sumber Daya Komputasi**

Model seperti FastText dirancang untuk bekerja dengan dataset yang cukup besar tanpa memerlukan sumber daya komputasi yang berat seperti GPU. Tetapi, menggunakan CPU untuk dataset dengan 119,211 baris membutuhkan waktu lebih lama dibandingkan menggunakan GPU. Waktu yang dibutuhkan untuk menjalankan 10 epoch yaitu 510.57 detik dengan menggunakan CPU. Penggunaan GPU memungkinkan untuk lebih mempersingkat waktu yang dibutuhkan.

- **Generalisasi**

Dengan 10 epoch, model memiliki peluang yang cukup untuk mempelajari pola dari data. Dengan akurasi yang didapatkan yaitu 73.613%, model yang dibangun sudah memiliki performa yang cukup baik. Namun, masih memiliki ruang untuk dilakukan perbaikan. Salah satu langkah yang bisa dilakukan yaitu melakukan preprocessing dengan baik sehingga model dapat lebih mampu menangkap pola tanpa adanya gangguan. Selain itu bisa dilakukan dengan mencoba berbagai macam parameter yang akan digunakan.

4. Transformer

```
Epoch 1/20, Loss: 0.7027, Test Accuracy: 0.4800
Epoch 2/20, Loss: 0.7038, Test Accuracy: 0.4800
Epoch 3/20, Loss: 0.6976, Test Accuracy: 0.4800
Epoch 4/20, Loss: 0.7138, Test Accuracy: 0.5200
Epoch 5/20, Loss: 0.7070, Test Accuracy: 0.4800
Epoch 6/20, Loss: 0.6991, Test Accuracy: 0.4800
Epoch 7/20, Loss: 0.7011, Test Accuracy: 0.5200
Epoch 8/20, Loss: 0.6991, Test Accuracy: 0.4800
Epoch 9/20, Loss: 0.7139, Test Accuracy: 0.4800
Epoch 10/20, Loss: 0.7068, Test Accuracy: 0.5200
Epoch 11/20, Loss: 0.7064, Test Accuracy: 0.4800
Epoch 12/20, Loss: 0.7003, Test Accuracy: 0.5200
Epoch 13/20, Loss: 0.6964, Test Accuracy: 0.4800
Epoch 14/20, Loss: 0.7094, Test Accuracy: 0.5200
Epoch 15/20, Loss: 0.7145, Test Accuracy: 0.4800
Epoch 16/20, Loss: 0.6999, Test Accuracy: 0.5200
Epoch 17/20, Loss: 0.7054, Test Accuracy: 0.4800
Epoch 18/20, Loss: 0.6989, Test Accuracy: 0.5200
Epoch 19/20, Loss: 0.6964, Test Accuracy: 0.5200
Epoch 20/20, Loss: 0.6971, Test Accuracy: 0.4800
Final Test Accuracy: 0.4800
Total computation time: 39.34 seconds
```

Hasil Akurasi Model Transformer

Kode : [PBA-TRANSFORMER](#)

- **Dataset**

Hasil akurasi test stagnan antara 48% dan 52%, model mengalami underfitting. Hal ini dapat disebabkan oleh dataset yang terlalu kecil, distribusi label yang tidak seimbang, atau preprocessing data yang tidak optimal. Untuk mengatasi masalah ini, dapat ditingkatkan dengan ukuran dataset, menggunakan peningkatan data, atau memastikan distribusi label lebih seimbang. Menggunakan teknik preprocessing yang baik, seperti tokenisasi yang tepat, penghapusan stopwords, dan normalisasi teks, sangat penting. Dataset kurang cocok dengan model transformer.

- **Waktu**

Hasil waktu komputasi 39.34 second , membuat model membutuhkan lebih banyak epoch atau sumber daya komputasi. Waktu pelatihan akan lebih lambat dengan CPU dibandingkan dengan GPU.

- **Generalisasi**

```

import torch
import torch.nn as nn
import numpy as np
from databits import CreateModel
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score, accuracy_score

BATCH_SIZE = 32
SEQUENCE_LENGTH = 100
EPOCHS = 5
EMBED_DIM = 512
N_LAYERS = 2
DROPOUT_RATE = 0.1
NUM_CLASSES = len(np.unique(np.array(y_train)))
OPTIMIZER = torch.optim.Adam
LR = 0.001
LOSS = nn.CrossEntropyLoss

```

Dengan, variasi seperti ini maka didapatkan akurasi final 40%. Untuk meningkatkan akurasi bisa dilakukan variasi yang lebih baik terhadap layer, batch size dan parameter lainnya.

5. BERT

Epoch 1: Train Loss: 0.2477, Test Accuracy: 0.9576

Epoch 2: Train Loss: 0.1481, Test Accuracy: 0.9680

Hasil Akurasi Model BERT

Kode: <https://www.kaggle.com/code/jasminegeorgina/tugas-pba-glove-bert>

- **Dataset**

Hasil model BERT menunjukkan kinerja yang sangat baik pada tahap awal pelatihan, dengan akurasi test yang meningkat dari 95.76% pada epoch pertama menjadi 96.80% pada epoch kedua, serta penurunan train loss dari 0.2477 menjadi 0.1481. Meskipun hasil ini mencerminkan performa yang optimal, pertimbangan terhadap ukuran dataset tetap diperlukan. Dataset yang lebih besar berpotensi meningkatkan generalisasi model, khususnya untuk mencegah overfitting dan memastikan bahwa performa tetap konsisten pada data yang belum pernah dilihat sebelumnya.

- **Waktu**

Dalam hal waktu dan sumber daya komputasi, proses pelatihan selama 1 jam 32 menit untuk menyelesaikan 2 epoch (sekitar 46 menit per epoch) menunjukkan bahwa model BERT memerlukan waktu yang cukup signifikan. Untuk mencapai akurasi yang lebih tinggi, dibutuhkan lebih banyak epoch, yang tentunya akan meningkatkan beban komputasi. Oleh karena itu, penggunaan sumber daya yang lebih optimal, seperti akselerator campuran antara CPU dan GPU, dapat menjadi solusi untuk mempercepat proses pelatihan tanpa mengorbankan kinerja model.

- **Generalisasi**

Generalisasi, yang merujuk pada kemampuan model untuk beradaptasi pada data baru di luar sampel pelatihan, terlihat cukup baik dari tingginya akurasi pada data uji dengan ukuran **batch size 128** dan pemanfaatan embedding **GloVe**. Kombinasi ini menunjukkan bahwa model memiliki pondasi yang kuat untuk memahami representasi teks. Namun, untuk memastikan performa tetap stabil pada data baru, penting untuk terus memantau perbedaan antara **train loss** dan **test loss**. Jika terdapat

indikasi overfitting, langkah-langkah seperti regularisasi, augmentasi data, atau eksplorasi embedding lain yang lebih kontekstual dapat diterapkan guna meningkatkan generalisasi model pada berbagai skenario data.