Tugas 3

Mata Kuliah Web Mining

Model BERT untuk *Sentiment Analysis*

Disusun Oleh:

Cornelius Justin Satryo Hadi	2006529796
Gladys Nathania Banuarea	2006534556
M. Hanif Pramudya Zamzami	2006487566
Tulus Setiawan	2006568802

Universitas Indonesia
Departemen Matematika

2023

Soal

Buatlah model BERT terbaik untuk sentiment analysis dengan menggunakan data training Capres2014-1.1. Source code program dan dataset dapat diperoleh pada Google Classroom.

Beberapa aspek yang dapat dioptimalkan untuk dapat membangun model terbaik, antara lain adalah data cleaning dan *hyperparameter* tuning.

Tugas dikumpulkan dalam satu file PDF yang berisi: penjelasan apa saja yang dioptimasi dalam membangun model, serta akurasi model.

Pendahuluan

I. Penjelasan Implementasi pada Data

- Memuat data
    - Perihal menyesuaikan permasalahan masalah, kolom yang dimuat hanya kolom isi_tweet dan sentimen
- Menampilkan data
    - Menampilkan 5 baris akhir data
- Mengecek *Imbalanced Data*
- Cleaning nilai dari fitur sentimen
    - konversi nilai dari {-1,1} menjadi {0,1}
- Cleaning nilai dari fitur isi_text
    - konversi kata menjadi huruf kecil (lower case)
    - menghapus muatan *mention* (@...)
    - menghapus tanda pagar (#....) menjadi kata-kata saja
    - menghapus spasi putih yang berlebih
    - menghapus muatan tautan (www. atau https://)
    - menghapus karakter *unicode* (*, ^,$,&,dll)
- Memisahkan data menjadi data train dan data test
    - pembagian menjadi 80% data train dan 20% data test

II. Penjelasan Implementasi pada Model

Model BERT yang digunakan untuk *Sentiment Analysis* memiliki parameter-parameter pada BERT tokenizer sebagai berikut :

- Tokenisasi dilakukan pada teks yang sudah dibersihkan
- add_special_tokern = True (menambahkan special token yang belum ada di *vocabulary*)
- max_length = 128 (panjang maksimal)
- padding =’max_length’ (padding dilakukan ke panjang yang ditentukan oleh ‘max_length’)
- truncation = True (potong ke panjang maksimum yang ditentukan oleh argumen max_length)
- return_attention_mask = True (Akan mengembalikan ‘topeng perhatian’)
- return_tensors = ‘tf’ (mengembalikan masing-masing TensorFlow)

Jawaban Soal

1. **Penjelasan Optimasi Model**

   Optimasi yang dilakukan adalah optimasi yang bertujuan untuk mencari jenis *hidden layer* yang tepat setelah BERT *embedding* serta *hyperparameter* yang digunakan adalah yang paling optimal. Untuk mendapatkan optimasi yang diharapkan, kami mendefinisikan metode optimasi untuk mendapat model terbaik dari 4 kandidat arsitektur model serta parameter - parameternya sebagai berikut :

   - Model pertama, arsitektur terdiri dari ;

BERT *embedding*	*Hidden Layer*: *Dense*	*Output Layer*
	Kandidat *hyperparameter* - Jumlah *hidden layer* : 1 atau 2 - Banyak neuron tiap layer : 32 ⁓ 128 - Regularisasi L2 : 0.001 ⁓ 0.1	**Kandidat *hyperparameter*** Regularisasi L2 : 0.001 ⁓ 0.1
	***Hyperparameter* tetap** - Fungsi aktivasi : relu	***Hyperparameter* tetap** - Jumlah neuron : 1 - Fungsi aktivasi : sigmoid

   - Model kedua, arsitektur terdiri dari ;

BERT *embedding*	*Hidden Layer*: LSTM	Output Layer
	Kandidat *hyperparameter* - LSTM bidirectional : *True* atau *False* - Dimensi output vektor : 32 ⁓ 128 - Regularisasi L2 : 0.001 ⁓ 0.1	**Kandidat *hyperparameter*** Regularisasi L2 : 0.001 ⁓ 0.1
		***Hyperparameter* tetap** - Jumlah neuron : 1 - Fungsi aktivasi : sigmoid

   - Model ketiga, arsitektur terdiri dari ;

BERT *embedding*	*Hidden Layer* 1: LSTM	*Hidden Layer* 2: *Dense*	*Output Layer*
	Kandidat	**Kandidat**	**Kandidat**

	hyperparameter - LSTM Bidirectional : *True* atau *False* - Dimensi output vektor : 32 ⁓ 128 - Regularisasi L2 : 0.001 ⁓ 0.1	***hyperparameter*** - Banyak neuron tiap layer : 32 ⁓ 128 - Regularisasi L2 : 0.001 ⁓ 0.1	***hyperparameter*** Regularisasi L2 : 0.001 ⁓ 0.1
		***Hyperparameter* tetap** - Fungsi aktivasi : relu - Jumlah *layer*: 1	***Hyperparameter* tetap** - Jumlah neuron : 1 - Fungsi aktivasi : sigmoid

- Model keempat, arsitektur terdiri dari ;

BERT *embedding*	Output Layer
	Kandidat *hyperparameter* - Regularisasi L2 : 0.001 ⁓ 0.1
	***Hyperparameter* tetap** - Jumlah neuron : 1 - Fungsi aktivasi : sigmoid

Sebelum proses *fitting* pada masing-masing model ke data *training*, beberapa *hyperparameter* yang dioptimasi pada perintah *compile* adalah sebagai berikut:

Kandidat *hyperparameter* Learning rate : 0.001 ⁓ 0.05
***Hyperparameter* tetap** Optimizer : Adam Loss : binary_crossentropy Metrics : accuracy

Selanjutnya dengan menggunakan ***Bayesian Search***, didapat arsitektur model terbaik dengan parameter paling optimalnya sebagai berikut :

BERT *embedding*	*Hidden Layer*: LSTM	Output Layer
	***Hyperparameter* paling optimal** LSTM Bidirectional : *True*	***Hyperparameter* paling optimal**

	Dimensi output vektor : 127 Regularisasi L2 : 0.00322	Regularisasi L2 : 0.06278
		***Hyperparameter* tetap** Jumlah neuron : 1 Fungsi aktivasi : sigmoid

serta, *hyperparameter* paling optimal pada perintah *compile* adalah sebagai berikut :

***Hyperparameter* paling optimal** Learning rate : 0.00553
***Hyperparameter* tetap** Optimizer : Adam Loss : binary_crossentropy Metrics : accuracy

Proses *hyperparameter tuning* dilakukan menggunakan 50 *epochs* dan *early stopping* dengan *patience* 5 untuk masing-masing kandidat arsitektur model dan *hyperparameter*.

Dengan arsitektur model dan *hyperparameter* teroptimal di atas menggunakan 50 *epochs* dan *early stopping* dengan *patience* 5, diperoleh metrik akurasi sebesar 0.8514.

Selanjutnya dilakukan proses *training* kembali menggunakan 100 *epochs* dan *early stopping* dengan *patience* 20 yang diharapkan dapat meningkatkan akurasi model.

2. **Hasil akurasi model**
   Setelah dilakukan proses *training* kembali menggunakan 100 *epochs* dan *early stopping* dengan *patience* 20, diperoleh metrik sebagai berikut:
   - **Loss** (binary cross entropy) : 0.5656
   - **Akurasi** : 0.8806

3. **Tautan Pengerjaan Model BERT pada Google Colab**
   https://colab.research.google.com/drive/1L_neWXsyMOKcFa27a1i9T7n2ChpldpXi?usp=sharing