UNIVERSITAS GUNADARMA

PRAKTIKUM KECERDASAN ARTIFICIAL

MANUAL BOOK

“Model Sentimen Analisis terhadap ulasan film”

Nama : Muhamad Priasmoro

NPM : 50421874

Kelas : 3IA11

Fakultas : Teknologi Industri

Jurusan : Informatika

PJ : Qonita Abdah

Ditulis Guna Melengkapi Sebagian Syarat

Praktikum Kecerdasan Artificial

Universitas Gunadarma

2023

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

AI atau Kecerdasan Buatan telah menjadi telah menjadi hal yang nyata dan penting di 2 tahun terakhir ini, salah satunya adalah Machine Learning. Salah satu teknologi Machine Learning yang cukup banyak dipakai adalah Natural Language Processing (NLP) . Natural Language Processing (NLP) adalah sebuah teknologi machine learning yang memberi komputer kemampuan untuk menginterpretasikan, memanipulasi, dan memahami bahasa manusia. Banyak organisasi dewasa ini memiliki begitu banyak data suara dan teks dari berbagai saluran komunikasi seperti email, pesan teks, umpan berita media sosial, video, audio, dan banyak lagi. Mereka menggunakan perangkat lunak NLP untuk memproses data ini secara otomatis, menganalisis maksud atau sentimen dalam pesan, dan merespons komunikasi manusia dalam waktu nyata.

Pada Manual Book ini Penulis akan mencoba untuk membuat salah satu pengaplikasian Natural Language Processing Yaitu Analisis Sentimen dengan menggunakan dataset ulasan film dari IMDB dan metode TensorFlow.

1.2 Tujuan

1 . Menjelaskan Apa itu Natural Language Processing

2. Menjelaskan Apa itu sentiment analisis

3. Memaparkan dan menjelaskan kode dan output sentiment analisis yang telah dibuat.

BAB II

PEMBAHASAN

2.1 Natural Language Processing

Natural Language Processing atau biasa disingkat NLP merupakan salah satu bidang ilmu komputer yang mempelajari interaksi computer dengan Bahasa yang digunakan secara umum dalam kehidupan sehari-hari. NLP mempelajari pengembangan teknik yang bertujuan bagaimana komputer memahami bahasa alami manusia. Bahasa alami yang digunakan oleh manusia dari berbagai negara akan memiliki perbedaan dalam bentuk penulisan dan pengucapan (Raharjo & Hartati, 2014).

Menurut (Lisangan, 2013) Natural Language Processing (NLP) dapat didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Untuk proses komputasi, bahasa harus direpresentasikan sebagai rangkaian simbol yang memenuhi aturan tertentu. Secara sederhana, NLP adalah mencoba untuk membuat komputer dapat mengerti perintah-perintah yang ditulis dalam standar bahasa manusia.

bidang-bidang di NLP

Menurut Pustejovky dan Stubbs (2012) dalam (Amelia, 2020) menjelaskan bahwa ada beberapa area utama penulisan pada bidang NLP, yaitu :

1. Question Answering System (QAS)

Suatu kemampuan dari komputer untuk menjawab beberapa pertanyaan yang diajukan oleh user. Dimulai dari memasukkan keyword ke dalam browser pencarian dengan QAS maka user dapat langsung bertanya sesuai bahasa natural yang digunakan, baik dalam bahasa Inggris, bahasa Mandarin, ataupun dalam bahasa Indonesia.

2. Summarization

Summarization merupakan pembuatan ringkasan dari beberapa kumpulan konten dokumen atau email. User dapat dibantu untuk melakukan konversi dokumen teks yang besar ke dalam bentuk slide presentasi yang efisien dengan menggunakan ini.

3. Machine Translation

Suatu produk yang menghasilkan aplikasi untuk memahami bahasa manusia dan menterjemahkannya ke dalam bahasa lain. Salah satu yang termasuk dalam aplikasi ini adalah Google Translate yang tujuannya mampu dalam menerjemahkan bahasa.

4. Speech Recognition

Pada bidang ini merupakan cabang ilmu dari NLP yang cukup rumit. Proses pembangunan model agar digunakan komputer atau telepon untuk mengenali bahasa yang diucapkan telah banyak dikerjakan. Bahasa yang digunakan berupa pertanyaan dan perintah.

5. Document Classification

Bidang penulisan NLP yang paling banyak digunakan dan paling sukses adalah Document Classification. Pekerjaan yang dilakukan oleh aplikasi ini adalah menentukan dimana tempat terbaik suatu dokumen yang baru di input ke dalam sistem. Aplikasi ini sangat berguna pada aplikasi spam filtering, news article classification, dan review pada suatu produk/aplikasi pada website.

2.2 Sentimen Analisis

Analisis sentimen adalah proses menganalisis teks digital untuk menentukan apakah nada emosional pesan tersebut positif, negatif, atau netral. Saat ini, perusahaan memiliki data teks dalam volume besar seperti email, transkrip obrolan dukungan pelanggan, komentar media sosial, dan ulasan. Alat analisis sentimen dapat memindai teks ini untuk secara otomatis menentukan sikap penulis terhadap suatu topik. Perusahaan menggunakan wawasan dari analisis sentimen untuk meningkatkan mutu layanan pelanggan dan meningkatkan reputasi merek.

Penggunaan analisis sentimen

Bisnis menggunakan analisis sentimen untuk memperoleh kecerdasan dan menyusun rencana yang dapat ditindaklanjuti di berbagai bidang.

- Meningkatkan mutu layanan pelanggan

Tim dukungan pelanggan menggunakan alat analisis sentimen untuk mempersonalisasi respons berdasarkan nuansa percakapan. Hal-hal yang mendesak terlihat oleh *chatbot* berbasis kecerdasan buatan (AI) dengan kemampuan analisis sentimen dan dieskalasi ke staf dukungan.

- Pemantauan merek

Organisasi terus memantau penyebutan dan obrolan seputar merek mereka di media sosial, forum, blog, artikel berita, dan di ruang digital lainnya. Teknologi analisis sentimen memungkinkan tim hubungan masyarakat menyadari cerita yang sedang berlangsung terkait. Tim dapat mengevaluasi nuansa yang mendasari untuk menangani keluhan atau memanfaatkan tren positif.

- Riset pasar

Sistem analisis sentimen membantu bisnis meningkatkan penawaran produk mereka dengan mempelajari apa yang berhasil dan apa yang tidak. Tenaga pemasaran dapat menganalisis komentar di situs ulasan *online*, jawaban survei, dan posting media sosia untuk mendapatkan wawasan yang lebih dalam tentang fitur produk tertentu. Mereka menyampaikan temuan tersebut kepada rekayasawan produk yang akan membuat inovasi sesuai wawasan tersebut.

Cara Kerja

Analisis sentimen adalah aplikasi teknologi pemrosesan bahasa alami (NLP) yang melatih perangkat lunak komputer untuk memahami teks dengan cara yang mirip dengan manusia. Analisis biasanya melewati beberapa tahap sebelum memberikan hasil akhir.

Prapemrosesan

Selama tahap prapemrosesan, analisis sentimen mengidentifikasi kata kunci untuk menyoroti pesan inti teks.

- Tokenisasi memecah kalimat menjadi beberapa elemen atau token.
- Lematisasi mengonversi kata-kata menjadi bentuk akarnya. Misalnya, bentuk akar *am* adalah *be*.
- Penghapusan kata henti memfilter kata-kata yang tidak menambah nilai bermakna ke kalimat. Misalnya, *with*, *for*, *at*, dan *of* adalah kata henti.

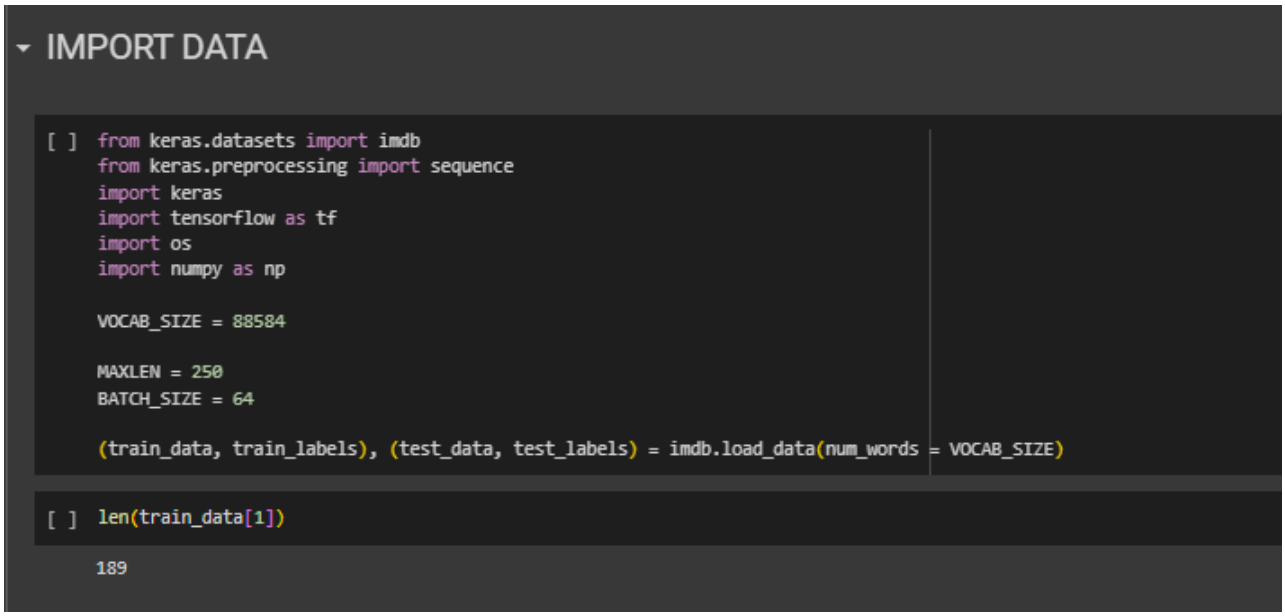
Analisis kata kunci

Teknologi NLP lebih lanjut menganalisis kata kunci yang diekstraksi dan memberi kata tersebut skor sentimen. Skor sentimen adalah skala pengukuran yang menunjukkan elemen emosional dalam sistem analisis sentimen. Ini memberikan persepsi yang terkait dengan emosi yang diekspresikan dalam teks untuk tujuan analitis. Misalnya, peneliti menggunakan 10 untuk merepresentasikan kepuasan dan 0 untuk kekecewaan saat menganalisis ulasan pelanggan.

BAB III

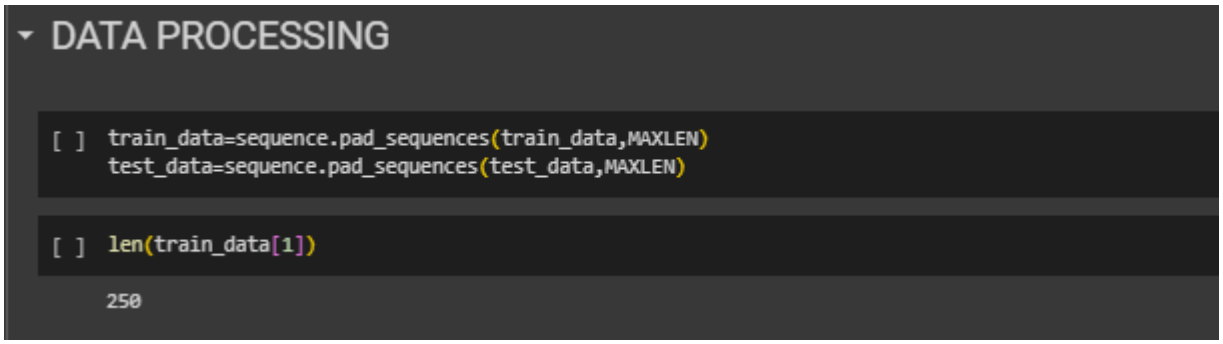
ANALISA DAN PERANCANGAN

Kode dan Penjelasan :



- ***from keras.datasets import imdb***: Mengimpor dataset IMDb dari modul keras.datasets.
- ***from keras.preprocessing import sequence***: Mengimpor fungsi sequence.pad_sequences() yang digunakan untuk mem-pad data ulasan agar memiliki panjang yang sama.
- ***import keras***: Mengimpor modul keras yang berisi berbagai fungsi dan kelas untuk membangun dan melatih model pembelajaran mesin.
- ***import tensorflow as tf***: Mengimpor modul tensorflow yang merupakan backend untuk keras.
- ***import os***: Mengimpor modul os yang menyediakan fungsi untuk berinteraksi dengan sistem operasi.
- ***import numpy as np***: Mengimpor modul numpy yang menyediakan fungsi untuk operasi matematika dan manipulasi array.
- ukuran kosakata ditetapkan sebagai 88584.

- panjang maksimum ulasan film ditetapkan sebagai 250 dan ukuran batch ditetapkan sebagai 64.
- ***Fungsi imdb.load_data()*** memuat data dataset IMDb dan mengembalikan empat variabel:
- ***train_data***: Data ulasan film latih yang disandikan sebagai urutan bilangan bulat.
- ***train_labels***: Label ulasan film latih, yang menunjukkan apakah ulasan film itu positif atau negatif.
- ***test_data***: Data ulasan film uji yang disandikan sebagai urutan bilangan bulat.
- ***test_labels***: Label ulasan film uji, yang menunjukkan apakah ulasan film itu positif atau negatif.
- ***Kode len(train_data[1])*** digunakan untuk mengetahui panjang ulasan film pertama dalam data latih. Panjang ulasan film dalam dataset IMDb bervariasi, tetapi panjang maksimum ulasan film adalah 250 kata.



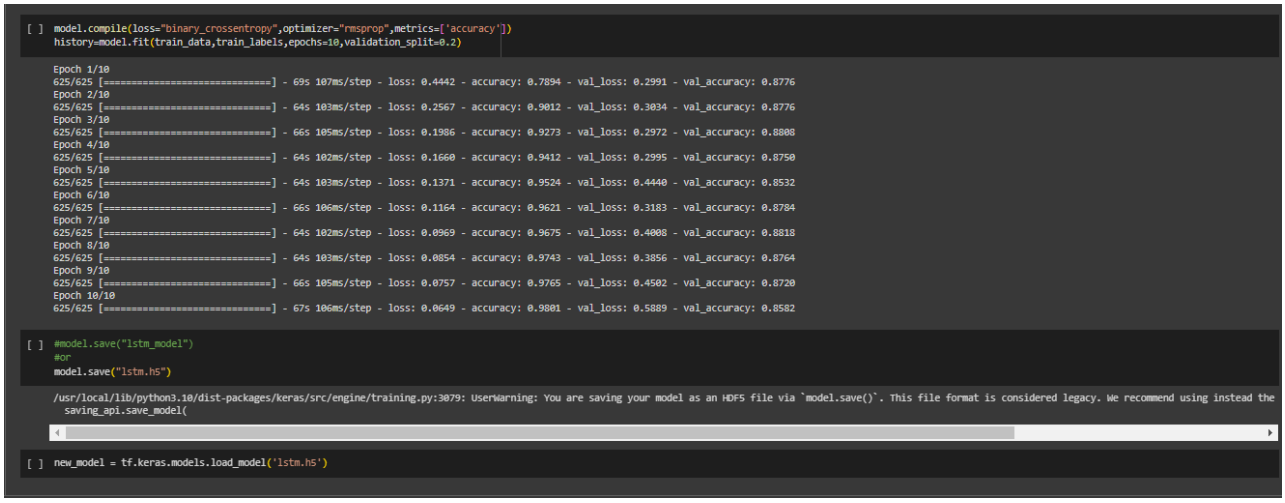
- ***`train_data`***: Data ulasan film latih.
- ***`maxlen`***: Panjang maksimum ulasan film. Dalam kode di atas, maxlen ditetapkan sebagai 250.
- ***`padding`***: Jenis padding yang digunakan. Dalam kode di atas, padding ditetapkan sebagai 'post', yang berarti bahwa padding akan ditambahkan di akhir ulasan film.
- Fungsi ***`sequence.pad_sequences()`*** akan mem-pad ulasan film yang lebih pendek dari 250 kata dengan menambahkan angka 0 di akhir ulasan film. Sedangkan ulasan film yang lebih panjang dari 250 kata akan dipotong menjadi 250 kata.

MEMBANGUN MODEL

```
[ ] model=tf.keras.Sequential([
        tf.keras.layers.Embedding(VOCAB_SIZE,32),
        tf.keras.layers.LSTM(32),
        tf.keras.layers.Dense(1,activation='sigmoid')
    ])
```

```
[ ] model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, None, 32)          2834688

 lstm_1 (LSTM)               (None, 32)                8320

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 2843041 (10.85 MB)
Trainable params: 2843041 (10.85 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



```
[ ] model.compile(loss="binary_crossentropy",optimizer="rmsprop",metrics=['accuracy'])
    history=model.fit(train_data,train_labels,epochs=10,validation_split=0.2)
```

```
Epoch 1/10
625/625 [==============================] - 69s 107ms/step - loss: 0.4442 - accuracy: 0.7894 - val_loss: 0.2991 - val_accuracy: 0.8776
Epoch 2/10
625/625 [==============================] - 64s 103ms/step - loss: 0.2567 - accuracy: 0.9012 - val_loss: 0.3034 - val_accuracy: 0.8776
Epoch 3/10
625/625 [==============================] - 66s 105ms/step - loss: 0.1986 - accuracy: 0.9273 - val_loss: 0.2972 - val_accuracy: 0.8808
Epoch 4/10
625/625 [==============================] - 64s 102ms/step - loss: 0.1660 - accuracy: 0.9412 - val_loss: 0.2995 - val_accuracy: 0.8750
Epoch 5/10
625/625 [==============================] - 64s 103ms/step - loss: 0.1371 - accuracy: 0.9524 - val_loss: 0.4440 - val_accuracy: 0.8532
Epoch 6/10
625/625 [==============================] - 66s 106ms/step - loss: 0.1164 - accuracy: 0.9621 - val_loss: 0.3183 - val_accuracy: 0.8784
Epoch 7/10
625/625 [==============================] - 64s 102ms/step - loss: 0.0969 - accuracy: 0.9675 - val_loss: 0.4008 - val_accuracy: 0.8818
Epoch 8/10
625/625 [==============================] - 64s 103ms/step - loss: 0.0854 - accuracy: 0.9743 - val_loss: 0.3856 - val_accuracy: 0.8764
Epoch 9/10
625/625 [==============================] - 66s 105ms/step - loss: 0.0757 - accuracy: 0.9765 - val_loss: 0.4502 - val_accuracy: 0.8720
Epoch 10/10
625/625 [==============================] - 67s 106ms/step - loss: 0.0649 - accuracy: 0.9801 - val_loss: 0.5889 - val_accuracy: 0.8582
```

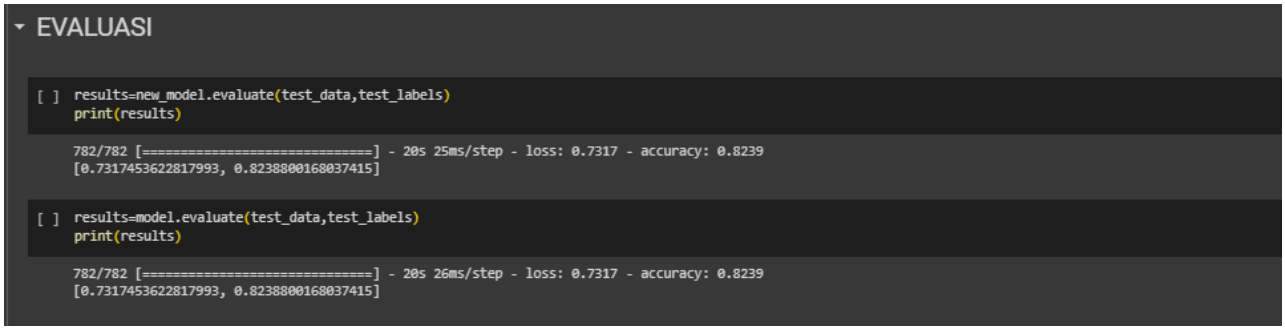
```
[ ] #model.save("lstm_model")
    #or
    model.save("lstm.h5")
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the
  saving_api.save_model(
```

```
[ ] new_model = tf.keras.models.load_model('lstm.h5')
```

- ***model=tf.keras.Sequential([tf.keras.layers.Embedding(VOCAB_SIZE,32),tf.keras.layers.LSTM(32),tf.keras.layers.Dense(1,activation='sigmoid')])*** digunakan untuk membangun model klasifikasi sentimen ulasan film. Model ini terdiri dari tiga lapisan:
  - Lapisan embedding: Lapisan embedding mengubah urutan bilangan bulat yang mewakili ulasan film menjadi vektor numerik.
  - Lapisan LSTM: Lapisan LSTM memproses vektor numerik ini untuk mengekstrak fitur yang relevan dari ulasan film.
  - Lapisan dense: Lapisan dense memprediksi sentimen ulasan film berdasarkan fitur yang diekstrak oleh lapisan LSTM.

- ***model.compile(loss="binary_crossentropy",optimizer="rmsprop",metrics=['accuracy'])*** digunakan untuk mengkompilasi model. Kompilasi model melibatkan menentukan fungsi loss yang akan digunakan untuk melatih model, optimizer yang akan digunakan untuk memperbarui parameter model, dan metrik yang akan digunakan untuk mengevaluasi kinerja model.
- ***history=model.fit(train_data,train_labels,epochs=10,validation_split=0.2)*** digunakan untuk melatih model. Pelatihan model melibatkan penyajian data ulasan film latih ke model dan memperbarui parameter model sehingga model dapat memprediksi sentimen ulasan film dengan akurasi yang tinggi. Parameter epochs menentukan jumlah iterasi yang akan dilakukan selama pelatihan model. Parameter validation_split menentukan proporsi data latih yang akan digunakan sebagai data validasi. Data validasi digunakan untuk mengevaluasi kinerja model selama pelatihan model.
- ***model.save("lstm.h5")*** digunakan untuk menyimpan model yang telah dilatih ke file lstm.h5. Fungsi ini memungkinkan model untuk dimuat ke dalam program lain untuk digunakan nanti.
- ***new_model = tf.keras.models.load_model('lstm.h5')*** digunakan untuk memuat model yang telah dilatih dari file lstm.h5.



- Kode ***results=new_model.evaluate(test_data,test_labels)*** dan ***results=model.evaluate(test_data,test_labels)*** digunakan untuk mengevaluasi kinerja model klasifikasi sentimen ulasan film pada data uji. Nilai loss dan akurasi yang dicetak ke konsol dapat digunakan untuk menilai kinerja model dan untuk membandingkan kinerja model yang berbeda.

MEMBUAT PREDIKSI

```
[ ]  word_index=imdb.get_word_index()
```

```
[ ]  for i in range(10):
         print(list(word_index.keys())[i],':',list(word_index.values())[i])
```

```
fawn : 34701
tsukino : 52006
nunnery : 52007
sonja : 16816
vani : 63951
woods : 1408
spiders : 16115
hanging : 2345
woody : 2289
trawling : 52008
```

```
[ ]  def encode_text(text):
         tokens=keras.preprocessing.text.text_to_word_sequence(text)
         tokens=[word_index[word] if word in word_index else 0 for word in tokens]
         return sequence.pad_sequences([tokens],MAXLEN)[0]
```

```
[ ]  text="that movie was good"
     encoded=encode_text(text)
     print(encoded)
```

```
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0  0  0  0  0 12 17 13 75]
```

```
[ ]  # Decode function that converts itegers to text

     reverse_word_index={value:key for (key,value) in word_index.items()}

     def decode_integers(integers):
         PAD=0
         text=""
         for num in integers:
             if num!=PAD:
                 text+=reverse_word_index[num] +" "

         return text[:-1]

     print(decode_integers(encoded))
```

```
that movie was bad
```

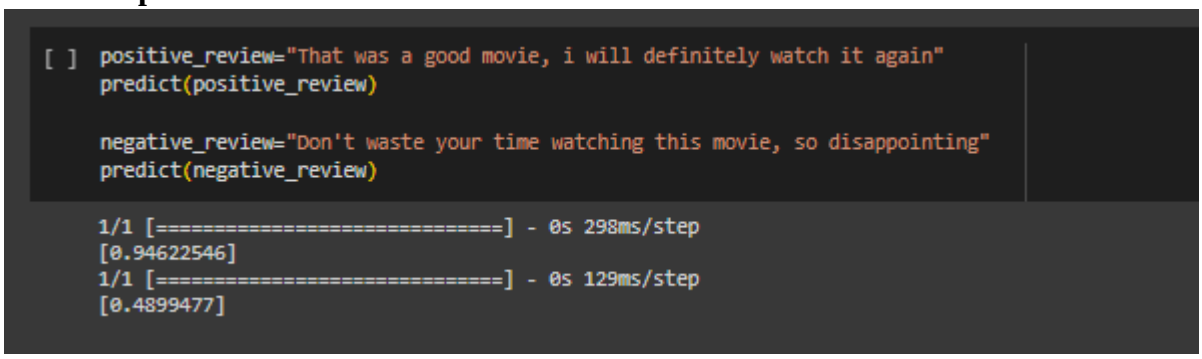
```
[ ]  # Decode function that converts itegers to text

     reverse_word_index={value:key for (key,value) in word_index.items()}

     def decode_integers(integers):
         PAD=0
         text=""
         for num in integers:
             if num!=PAD:
                 text+=reverse_word_index[num] +" "

         return text[:-1]

     print(decode_integers(encoded))
```

```
     that movie was bad
```

```
[ ]  def predict(text):
         encoded_text=encode_text(text)
         pred=encoded_text.reshape(1,250) #converting vector to 2d
         result=model.predict(pred)
         print(result[0])
```

- Kode ***word_index=imdb.get_word_index()*** digunakan untuk mendapatkan kamus kata-kata yang digunakan dalam dataset IMDb, di mana setiap kata dipetakan ke sebuah indeks.
- Kode ***def encode_text(text)*****:** mendefinisikan fungsi untuk menyandikan ulasan film menjadi urutan bilangan bulat. Fungsi ini bekerja dengan cara memecah ulasan film menjadi token, kemudian mencari indeks token tersebut dalam kamus kata-kata. Jika token tidak ditemukan dalam kamus kata-kata, maka token tersebut akan diganti dengan 0.
- Kode ***encoded=encode_text(text)*** menyandikan ulasan film text menjadi urutan bilangan bulat.
- Kode ***def decode_integers(integers)*****:** mendefinisikan fungsi untuk mendekode urutan bilangan bulat menjadi ulasan film. Fungsi ini bekerja dengan cara mencari token yang sesuai dengan setiap bilangan bulat dalam urutan bilangan bulat tersebut.
- Kode ***print(decode_integers(encoded))*** mendekode urutan bilangan bulat encoded menjadi ulasan film.
- Kode ***def predict(text)*****:** mendefinisikan fungsi untuk memprediksi sentimen ulasan film. Fungsi ini bekerja dengan cara menyandikan ulasan film text menjadi urutan bilangan bulat, kemudian memprediksi sentimen ulasan film tersebut menggunakan model klasifikasi sentimen ulasan film.

Hasil/Output



```
[ ]  positive_review="That was a good movie, i will definitely watch it again"
     predict(positive_review)

     negative_review="Don't waste your time watching this movie, so disappointing"
     predict(negative_review)
```

```
1/1 [==============================] - 0s 298ms/step
[0.94622546]
1/1 [==============================] - 0s 129ms/step
[0.4899477]
```

- Kode ***positive_review="That was a good movie, i will definitely watch it again" predict(positive_review)*** memprediksi sentimen ulasan film positive_review dan memberikan Output dalam bentuk angka yaitu 0.9 dari 1.0
- Kode ***negative_review="Don't waste your time watching this movie, so disappointing" predict(negative_review)*** memprediksi sentimen ulasan film negative_review dan memberikan Output dalam bentuk angka yaitu 0.4 dari 1.0

BAB IV

PENUTUP

4.1 Kesimpulan

Manual Book ini merupakan implementasi model sentimen analisis dengan menggunakan Teknik Natural Language Processing untuk menganalisis ulasan film dari dataset imdb. Metode yang digunakan adalah TensorFlow dan model mampu menghasilkan prediksi sentimen untuk teks input, yang berguna untuk menilai apakah suatu ulasan dianggap positif atau negative dalam bentuk angka. Dengan demikian, makalah ini memberikan gambaran umum tentang penerapan NLP dalam analisis sentimen untuk ulasan film.

4.2 Saran

1. Pembuatan model harus menggunakan optimizer yang lebih cepat
2. Mempertimbangkan penggunaan callback, seperti ModelCheckpoint atau EarlyStopping, untuk menyimpan model terbaik.
3. Menambahkan komentar di beberapa bagian penting dari kode untuk menjelaskan logika atau tujuan dari langkah-langkah tertentu