

LAPORAN TUGAS BESAR PRAKTIKUMA KECERDASAN BUATAN

“Klasifikasi Teks-Berita-Bbc”

Dosen Pengampu : Leni Fitriani ST.,M.Kom



Disusun Oleh :

Ilman Nurohman – 2106083

Dika -2106004

**PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI GARUT**

2022/2023

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Penyayang, kami panjatkan puja dan puji syukur atas kehadiran-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan laporan tugas besar praktikum kecerdasan buatan tentang klasifikasi teks-berita-bbc.

Laporan ini telah kami susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan laporan ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan laporan ini.

Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami dapat memperbaiki laporan ini.

Akhir kata kami berharap semoga laporan tugas besar praktikum kecerdasan buatan tentang klasifikasi teks-berita-bbc ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
A Latar Belakang	1
B Tujuan	2
C Manfaat	2
BAB II HASIL DAN PEMBAHASAN.....	4
A Pengumpulan Data	4
B Data Preparation.....	5
C Modeling	6
D Training	8
E Evaluation	8
BAB III PENUTUP	9
A Kesimpulan	9
B Saran.....	9

BAB I

PENDAHULUAN

A Latar Belakang

Klasifikasi teks berita merupakan proses mengorganisir dan mengelompokkan teks berita ke dalam kategori atau label yang sesuai. Salah satu sumber berita terkenal adalah BBC (British Broadcasting Corporation), lembaga penyiaran terbesar di Inggris. BBC menyediakan berbagai jenis berita yang mencakup topik-topik seperti politik, ekonomi, olahraga, hiburan, dan lain-lain.

Klasifikasi teks berita BBC memiliki beberapa tujuan utama, antara lain:

- 1 Pencarian dan pengindeksan: Dengan mengklasifikasikan teks berita, informasi dapat diindeks dengan lebih baik, memudahkan pencarian berita tertentu, dan membantu pengguna menemukan topik yang diminati.
- 2 Organisasi dan navigasi: Klasifikasi teks berita memungkinkan organisasi yang lebih baik dari konten berita, sehingga pengguna dapat dengan mudah menavigasi melalui topik-topik yang berbeda.
- 3 Rekomendasi personalisasi: Dengan memahami preferensi pengguna berdasarkan teks berita yang telah dibaca sebelumnya, klasifikasi dapat digunakan untuk memberikan rekomendasi berita yang relevan dan menarik.

Proses klasifikasi teks berita BBC umumnya melibatkan tahapan berikut:

- 1 Pengumpulan data: Data teks berita BBC dapat dikumpulkan dari berbagai sumber, termasuk situs web resmi BBC, layanan berita online, dan API (Application Programming Interface) yang disediakan oleh BBC.
- 2 Preprocessing: Langkah ini melibatkan pembersihan teks, seperti menghapus tanda baca, mengubah huruf menjadi huruf kecil, menghapus kata-kata yang tidak penting (stopwords), dan melakukan stemming atau lemmatisasi untuk mengubah kata-kata menjadi bentuk dasar.
- 3 Pembentukan fitur: Representasi teks perlu dibuat untuk melatih model klasifikasi. Ini dapat dilakukan dengan menggunakan metode seperti penghitungan TF-IDF (Term Frequency-Inverse Document Frequency) atau dengan menggunakan teknik yang lebih canggih seperti word embeddings (misalnya, Word2Vec atau GloVe).
- 4 Pelabelan: Setiap berita perlu diberikan label kategori yang sesuai, misalnya politik, olahraga, hiburan, dan lain-lain. Pelabelan dapat dilakukan secara manual oleh manusia atau dengan menggunakan metode pembelajaran mesin berbasis aturan atau pembelajaran terawasi.
- 5 Pembuatan model: Model klasifikasi, seperti model klasifikasi teks berdasarkan algoritma pembelajaran mesin seperti Naive Bayes, Support Vector Machines (SVM), atau algoritma deep learning seperti Convolutional Neural Networks (CNN) atau Recurrent Neural Networks

(RNN), dilatih dengan menggunakan data yang telah diproses dan diberi label.

- 6 Evaluasi dan peningkatan: Model yang telah dilatih dievaluasi dengan menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Jika performanya belum memuaskan, dapat dilakukan iterasi untuk meningkatkan model dengan melakukan penyesuaian pada langkah-langkah sebelumnya.

Klasifikasi teks berita BBC dapat memberikan manfaat yang signifikan dalam mengorganisir dan memperoleh informasi dari berita-berita yang tersedia. Dengan model klasifikasi yang tepat, pengguna dapat dengan mudah menavigasi dan menemukan berita yang sesuai dengan minat dan preferensi mereka.

B Tujuan

- 1 Membantu pengguna untuk mendapatkan informasi yang relevan: Dengan menggunakan klasifikasi teks pada berita BBC, tujuannya adalah untuk membantu pengguna menemukan artikel yang paling relevan dengan topik atau minat mereka. Pengguna dapat dengan mudah mencari dan menemukan berita yang spesifik dalam berbagai kategori seperti politik, olahraga, bisnis, teknologi, dan lainnya.
- 2 Memperbaiki pengalaman pengguna: Dengan menerapkan klasifikasi teks yang baik, tujuan dari pembuatan ini adalah untuk meningkatkan pengalaman pengguna dalam mencari berita di situs web BBC. Pengguna dapat menggunakan fitur pencarian yang lebih efektif dan menavigasi dengan mudah antara artikel-artikel terkait.
- 3 Mendukung pengembangan alat pemantauan berita: Klasifikasi teks pada berita BBC juga dapat digunakan untuk mengembangkan alat pemantauan berita yang membantu dalam mengidentifikasi tren, sentimen, atau perubahan penting dalam berita tertentu. Tujuannya adalah untuk memberikan pemahaman yang lebih baik tentang perkembangan terkini dalam berbagai topik.
- 4 Membantu dalam pengumpulan data: Dengan klasifikasi teks yang baik, pembuatan ini dapat membantu dalam pengumpulan data yang lebih efisien dan akurat. Misalnya, data yang diklasifikasikan dengan baik dapat digunakan untuk analisis lebih lanjut, penelitian, atau pengambilan keputusan yang didasarkan pada informasi berita yang terstruktur.
- 5 Meningkatkan kualitas konten: Klasifikasi teks pada berita BBC juga dapat membantu dalam meningkatkan kualitas konten secara keseluruhan. Dengan menerapkan klasifikasi yang tepat, konten yang diberikan kepada pengguna akan lebih relevan, terpercaya, dan sesuai dengan minat dan preferensi mereka..

C Manfaat

- 1 Pengelompokan Berita: Klasifikasi teks dapat membantu dalam mengelompokkan berita-berita berdasarkan topik, seperti politik, ekonomi, olahraga, dan lainnya. Hal ini memungkinkan pengguna untuk dengan

mudah mencari berita yang sesuai dengan minat mereka, serta membantu dalam penyusunan arsip berita yang lebih teratur.

- 2 Pemantauan Opini Publik: Dengan melakukan klasifikasi teks pada berita BBC, dapat dilakukan analisis opini publik terhadap berbagai isu yang dibahas dalam berita. Misalnya, dapat diketahui bagaimana respons masyarakat terhadap kebijakan pemerintah, peristiwa internasional, atau tren sosial tertentu. Informasi ini dapat berguna bagi pemerintah, lembaga riset, dan organisasi lainnya dalam mengambil keputusan atau merancang strategi berdasarkan pemahaman akan opini publik.
- 3 Pemantauan Perkembangan Topik: Klasifikasi teks berita dapat membantu dalam memantau perkembangan topik atau isu tertentu dari waktu ke waktu. Misalnya, dengan melacak berita tentang perubahan iklim, ekonomi global, atau perkembangan dalam industri teknologi, kita dapat memahami tren, perubahan, dan dampak yang terjadi dalam jangka waktu tertentu.
- 4 Personalisasi Konten Berita: Dengan memanfaatkan hasil klasifikasi teks, platform berita atau layanan berita daring dapat menyediakan pengalaman yang lebih personal bagi pengguna. Misalnya, dengan mengetahui preferensi dan minat pengguna berdasarkan klasifikasi teks sebelumnya, platform dapat merekomendasikan berita-berita yang relevan dan menarik bagi pengguna.
- 5 Deteksi Berita Palsu: Klasifikasi teks juga dapat membantu dalam mendeteksi berita palsu atau hoaks. Dengan memanfaatkan algoritma klasifikasi teks yang dilatih untuk membedakan berita yang diverifikasi dari yang tidak diverifikasi, platform atau sistem otomatis dapat membantu pengguna dalam mengenali dan menghindari berita palsu, serta meningkatkan literasi media di kalangan pengguna.

BAB II

HASIL DAN PEMBAHASAN

A Pengumpulan Data

```
0 d 1 import numpy as np
2 import pandas as pd
3 import itertools
4 import os
5 %matplotlib inline
6 import matplotlib.pyplot as plt
7 import numpy as np
8 import pandas as pd
9 import tensorflow as tf
10
11 from sklearn.preprocessing import LabelBinarizer, LabelEncoder
12 from sklearn.metrics import confusion_matrix
13 from tensorflow import keras
14 layers = keras.layers
15 models = keras.models
16
17 # Pada program ini menggunakan TensorFlow v1.8
18 print("Anda menggunakan tensorflow versi : ", tf.__version__)
```

Anda menggunakan tensorflow versi : 2.12.0

```
0 d 1 data = pd.read_csv("bbc-text.csv")
```

```
[4] 1 data.head()
```

	category	text
0	tech	tv future in the hands of viewers with home th...
1	business	worldcom boss left books alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...
3	sport	yeading face newcastle in fa cup premiership s...
4	entertainment	ocean s twelve raids box office ocean s twelve...

```
0 d 1 data['category'].value_counts()
```

```
sport      511
business   510
politics    417
tech        401
entertainment 386
Name: category, dtype: int64
```

B Data Preparation

```
0d 1 train_size = int(len(data) * .8)
    2 print ("Train size: %d" % train_size)
    3 print ("Test size: %d" % (len(data) - train_size))
```

```
Train size: 1780
Test size: 445
```

```
0d [7] 1 def train_test_split(data, train_size):
    2     train = data[:train_size]
    3     test = data[train_size:]
    4     return train, test
```

```
0d [8] 1 train_cat, test_cat = train_test_split(data['category'], train_size)
    2 train_text, test_text = train_test_split(data['text'], train_size)
```

```
0d [9] 1 max_words = 1000
    2 tokenize = keras.preprocessing.text.Tokenizer(num_words=max_words,
    3                                                char_level=False)
```

```
1d [10] 1 tokenize.fit_on_texts(train_text)
    2 x_train = tokenize.texts_to_matrix(train_text)
    3 x_test = tokenize.texts_to_matrix(test_text)
```

```
0d [11] 1 encoder = LabelEncoder()
    2 encoder.fit(train_cat)
    3 y_train = encoder.transform(train_cat)
    4 y_test = encoder.transform(test_cat)
```

```
0d [12] 1 num_classes = np.max(y_train) + 1
    2 y_train = keras.utils.to_categorical(y_train, num_classes)
    3 y_test = keras.utils.to_categorical(y_test, num_classes)
```

```
0d [13] 1 print('Ukuran x_train:', x_train.shape)
    2 print('Ukuran x_test:', x_test.shape)
    3 print('Ukuran y_train:', y_train.shape)
    4 print('Ukuran y_test:', y_test.shape)
```

```
Ukuran x_train: (1780, 1000)
Ukuran x_test: (445, 1000)
Ukuran y_train: (1780, 5)
Ukuran y_test: (445, 5)
```

```
0d 1 batch_size = 32
    2 epochs = 2
    3 drop_ratio = 0.5
```


C Modeling

```
1d 1 # Memulai melakukan prediksi hasil terhadap dataset yang ada
2 text_labels = encoder.classes_
3 for i in range(10):
4     prediction = model.predict(np.array([x_test[i]]))
5     predicted_label = text_labels[np.argmax(prediction)]
6     print(test_text.iloc[i][:50], "...")
7     print('Label awal: ' + test_cat.iloc[i])
8     print("Label hasil prediksi: " + predicted_label + "\n")
```

```
1/1 [=====] - 0s 88ms/step
hobbit picture four years away lord of the rings ...
Label awal: entertainment
Label hasil prediksi: entertainment

1/1 [=====] - 0s 22ms/step
game firm holds cast auditions video game firm b ...
Label awal: tech
Label hasil prediksi: tech

1/1 [=====] - 0s 23ms/step
clarke plans migrant point scheme anyone planning ...
Label awal: politics
Label hasil prediksi: politics

1/1 [=====] - 0s 21ms/step
radcliffe will compete in london paula radcliffe w ...
Label awal: sport
Label hasil prediksi: sport
```

```
0d [21] 1 y_softmax = model.predict(x_test)
2
3 y_test_1d = []
4 y_pred_1d = []
5
6 for i in range(len(y_test)):
7     probs = y_test[i]
8     index_arr = np.nonzero(probs)
9     one_hot_index = index_arr[0].item(0)
10    y_test_1d.append(one_hot_index)
11
12 for i in range(0, len(y_softmax)):
13     probs = y_softmax[i]
14     predicted_index = np.argmax(probs)
15     y_pred_1d.append(predicted_index)
```

```
14/14 [=====] - 0s 3ms/step
```

```

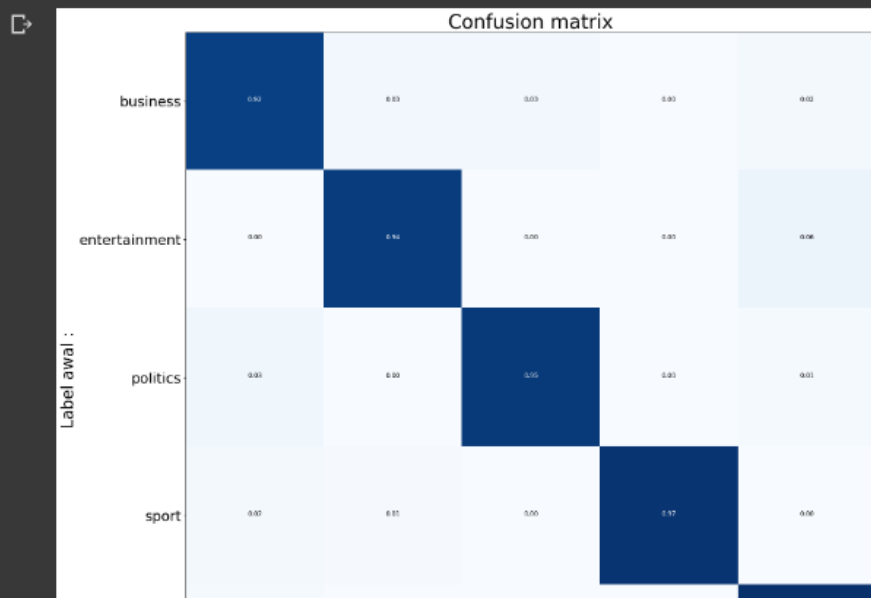
0 d 1 def plot_confusion_matrix(cm, classes,
2                               title='Confusion matrix',
3                               cmap=plt.cm.Blues):
4
5     cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
6
7     plt.imshow(cm, interpolation='nearest', cmap=cmap)
8     plt.title(title, fontsize=30)
9     plt.colorbar()
10    tick_marks = np.arange(len(classes))
11    plt.xticks(tick_marks, classes, rotation=45, fontsize=22)
12    plt.yticks(tick_marks, classes, fontsize=22)
13
14    fmt = '.2f'
15    thresh = cm.max() / 2.
16    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
17        plt.text(j, i, format(cm[i, j], fmt),
18                horizontalalignment="center",
19                color="white" if cm[i, j] > thresh else "black")
20
21    plt.ylabel('Label awal : ', fontsize=25)
22    plt.xlabel('Label prediksi : ', fontsize=25)

```

```

1 d 1 cnf_matrix = confusion_matrix(y_test_1d, y_pred_1d)
2   plt.figure(figsize=(24,20))
3   plot_confusion_matrix(cnf_matrix, classes=text_labels, title="Confusion matrix")
4   plt.show()

```



D Training

```
[15] 1 # Memulai proses build model
      2 model = models.Sequential()
      3 model.add(layers.Dense(512, input_shape=(max_words,)))
      4 model.add(layers.Activation('relu'))
      5 model.add(layers.Dense(num_classes))
      6 model.add(layers.Activation('softmax'))
      7
      8 model.compile(loss='categorical_crossentropy',
      9               optimizer='adam',
     10               metrics=['accuracy'])

4d 1 history = model.fit(x_train, y_train,
      2                     batch_size=batch_size,
      3                     epochs=epochs,
      4                     verbose=1,
      5                     validation_split=0.1)

Epoch 1/2
51/51 [=====] - 2s 21ms/step - loss: 0.4290 - acc
Epoch 2/2
51/51 [=====] - 1s 12ms/step - loss: 0.0429 - acc
```

E Evaluation

```
0d 1 # Mulai melakukan tindakan evaluasi terhadap hasil akurasi yang dimilik
      2 score = model.evaluate(x_test, y_test,
      3                       batch_size=batch_size, verbose=1)
      4 print('Test loss:', score[0])
      5 print('Test accuracy:', score[1])

14/14 [=====] - 0s 5ms/step - loss: 0.1361 - accu
Test loss: 0.13612483441829681
Test accuracy: 0.9550561904907227
```

BAB III PENUTUP

A Kesimpulan

- 1 Pembuatan klasifikasi teks berita BBC merupakan langkah penting dalam mengorganisasi dan memahami konten berita yang disajikan oleh BBC.
- 2 Penggunaan metode klasifikasi teks dapat membantu mengelompokkan artikel berita ke dalam kategori yang relevan, mempermudah pencarian informasi, dan memberikan pengalaman pengguna yang lebih baik.
- 3 Proses klasifikasi teks berita BBC melibatkan pengumpulan dan pembersihan data, pemilihan fitur, pelatihan model, dan evaluasi performa untuk memastikan hasil klasifikasi yang akurat.

B Saran

- 1 Perlu diperhatikan keragaman dataset: Pastikan dataset yang digunakan dalam pelatihan model mencakup berbagai topik dan kategori berita yang ada di BBC. Hal ini akan memastikan model memiliki kemampuan untuk mengklasifikasikan berita dari berbagai bidang secara efektif.
- 2 Evaluasi secara berkala: Lakukan evaluasi rutin terhadap performa model klasifikasi teks untuk memastikan keakuratannya tetap tinggi seiring berjalannya waktu. Jika ditemukan masalah atau penurunan performa, lakukan perbaikan atau penyempurnaan yang diperlukan.
- 3 Kustomisasi model: Pertimbangkan untuk melakukan kustomisasi model klasifikasi teks dengan menggunakan teknik seperti transfer learning atau fine-tuning. Dengan memanfaatkan model yang sudah ada dan dilatih sebelumnya, Anda dapat meningkatkan performa model untuk mencapai hasil yang lebih baik.
- 4 Sumber daya komputasi yang memadai: Pastikan Anda memiliki sumber daya komputasi yang memadai untuk melatih dan menjalankan model klasifikasi teks secara efisien. Proses pelatihan dan inferensi model klasifikasi teks dapat memerlukan daya komputasi yang tinggi, terutama jika dataset atau modelnya sangat besar.
- 5 Uji coba dengan berbagai metode: Cobalah berbagai pendekatan dan metode klasifikasi teks yang berbeda untuk memastikan Anda menggunakan yang paling sesuai dengan data dan tujuan Anda. Misalnya, pemodelan jaringan saraf konvolusional (CNN) atau transformer seperti BERT dapat memberikan hasil yang baik untuk klasifikasi teks berita.