Muhammad Andhika Ramadhan

AlphaZero

Perbandingan Logistic Regression dengan BERT untuk Analisis Sentimen Cyberbullying Instagram

Ujian Praktik Natural Language Processing – Orbit Future Academy

Latar Belakang, Tujuan dan Urgensi

- **Latar Belakang**
  - Hampir setiap platform media social memiliki fitur chat, komentar
  - Pada kali ini komentar terbagi menjadi 2, positive dan negative
    - Dimana negative cenderung menggunakan kata kasar
  - Dataset berasal dari platform Instagram.

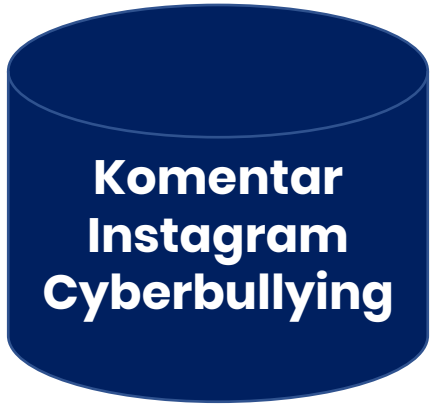
- **Tujuan**
  - Membuat dan membandingkan dua buah arsitektur untuk mengklasifikasikan jenis sentiment apa pada dataset dan data pengujian nanti

- **Urgensi**
  - Membuat sebuah fitur auto delete atau tidak menampilkan pesan apabila terdeteksi komentar yang memang tidak terpuji dan cenderung berbahasa kasar

Data, Variabel yang digunakan

Train: 80% || Val: 20%

320 || 80



Fitur

Fitur	Penjelasan
Instagram Comment	Mengenai komen-komen pada Instagram yang telah dikumpulkan

Sentimen	Instagram Comment Text
negative	<USERNAME> TOLOL!! Gak ada hubungan nya keguguran dgn pake hijab syar'i yg lo bilang bayi nya kepanasan didalem
negative	Geblek lo tata...cowo bgt dibela2in balikan...hadeww...ntar ditinggal lg nyalahin tuh cowo...padahal kitenya yg oon.
positive	Polossss...keren....ga neko2 kayaknya...jossss
positive	Yesssssss !!!!! Bagus bgt !!! ?????? Kids jaman now yg begini yg diksh jempolllll ????????

Target

Label	Penjelasan
Sentiment	Dibagi menjadi 2, positive dan negative

Preprocessing

Negative, Positive to 0,1

03 Text Preprocessing

In [41]:
```
1 data.replace(to_replace=['negative', 'positive'], value=[0,1], inplace=True)
2 data
```

Out[41]:

	Id	Sentiment	Instagram Comment Text
0	1	0	<USERNAME> TOLOL!! Gak ada hubungan nya kegug...
1	2	0	Geblek lo tata...cowo bgt dibela2in balikan.....
2	3	0	Kmrn termewek2 skr lengket lg duhhh kok labil ...
3	4	0	Intinya kalau kesel dengan ATT nya, gausah ke ...
4	5	0	hadewwwww permpuan itu lg!!!!sakit jiwa,knp ha...
...	...	...	...
395	396	1	Bangga sama suami yg selalu ingat istri disela...
396	397	1	Apaoun pekerjaannya yg penting halal u tuk men...
397	398	1	Gojek itu mayoritas pegangguran yang lama gak ...
398	399	1	<USERNAME> aslinya cantik dan ayu loh mbak kr...
399	400	1	<USERNAME> suami saya seumuran sama saya mba,...

Preprocessing

Normalizer

In [42]:
```
key_norm = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/dataset/key_norm.csv')
print(key_norm.head())
key_norm.shape
```

```
   _id      singkat       hasil
0    1         abis       habis
1    2       accent     tekanan
2    3       accept      terima
3    4     accident  kecelakaan
4    5  achievement    prestasi
```

Out[42]: (3720, 3)

```
import re

def text_preprocessing(text):
  text = text_normalize(text) # lemitisasi
  text = remove_stop_words(text)
  text = text.lower()                                  # Mengubah teks menjadi lower case
  text = re.sub(r'https?://\S+|www\.\S+', '', text) # Menghapus URL
  text = re.sub(r'[-+]?[0-9]+', '', text)              # Menghapus angka
  text = re.sub(r'[^\w\s]','', text)                   # Menghapus karakter tanda baca
  text = text.strip()                                  # Menghapus whitespaces
  return text
```

Preprocessing

Stop Words

```
from nltk.corpus import stopwords
from string import punctuation

plusStopword = ['<username>', '@w.lina2706']
sw_indo = stopwords.words('indonesian') + list(punctuation) + plusStopword

def remove_stop_words(text):
  clean_words = []
  text = text.split()
  for word in text:
      if word not in sw_indo:
          clean_words.append(word)
  return " ".join(clean_words)
```

Feature Extraction

Machine Learning

```
pipeline = Pipeline([
    ('prep', TfidfVectorizer(tokenizer=word_tokenize, stop_words=sw_indo)),
    ('algo', LogisticRegression(solver='lbfgs', n_jobs=-1, random_state=42))
])
```

BERT

In [52]:
```
# Tentukan pre-trained model yang akan digunakan untuk fine-tuning
# Daftar model dapat ditemukan pada https://huggingface.co

PRE_TRAINED_MODEL = 'indobenchmark/indobert-base-p2'  # https://huggingface.co/indobenchmark/indobert-base-p2
```

In [53]:
```
from transformers import BertTokenizer

bert_tokenizer = BertTokenizer.from_pretrained(PRE_TRAINED_MODEL)  # Load tokenizer dari pre-trained model
```

Model dan Parameter

Machine Learning

```
%%time

pipeline = Pipeline([
    ('prep', TfidfVectorizer(tokenizer=word_tokenize, stop_words=sw_indo)),
    ('algo', LogisticRegression(solver='lbfgs', n_jobs=-1, random_state=42))
])

parameter = {
    'algo__fit_intercept': [True, False],
    'algo__multi_class': ['ovr', 'multinomial'],
    'algo__C': [1.e-03, 1.e-02, 1.e-01, 1.e+00, 1.e+01, 1.e+02, 1.e+03]
}

# for i in range(2,12):
model = RandomizedSearchCV(pipeline, parameter, cv=5, n_iter=50, n_jobs=-1, verbose=1, random_state=42)
model.fit(X_train, y_train)
print(model.best_params_)
print(model.score(X_train, y_train), model.best_score_, model.score(X_test, y_test))
print(" ")
```

Model dan Parameter

BERT

```
BATCH_SIZE = 32
LEARNING_RATE = 5e-5
```

```
'''
BERT untuk tugas klasifikasi sequence (teks) dengan menambahkan linear layer di atas pooled output untuk pengklasifikasi
https://huggingface.co/docs/transformers/model_doc/bert#transformers.TFBertForSequenceClassification
'''

from transformers import TFBertForSequenceClassification

# Load model
bert_model = TFBertForSequenceClassification.from_pretrained(PRE_TRAINED_MODEL, num_labels=2)
```

```
Downloading:   0%|          | 0.00/656M [00:00<?, ?B/s]
```

```
All model checkpoint layers were used when initializing TFBertForSequenceClassification.

Some layers of TFBertForSequenceClassification were not initialized from the model checkpoint at indobenchmark/indobert-base-p2
and are newly initialized: ['classifier']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
# Tentukan optimizer dengan learning rate tertentu
# Paper aslinya menggunakan Adam Optimizer
optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)

# Karena tidak menggunakan one-hot vectors, sehingga loss function dapat menggunakan sparse categorical cross entropy
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metric = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')

# Compile model
bert_model.compile(optimizer=optimizer, loss=loss, metrics=[metric])
```

Model dan Parameter

Parameter	Value
Optimizer	Adam
Learning Rate	$5e^{-5}$
Loss	Categorical_crossentropy
Metrics	accuracy

Performa

Machine Learning

In [13]:
```
#Check performa model menggunakan classification_report
from sklearn.metrics import classification_report
y_pred_logreg = model.predict(X_test)
print(classification_report(y_test, y_pred_logreg))
```

```
              precision    recall  f1-score   support

           0       0.90      0.95      0.93        40
           1       0.95      0.90      0.92        40

    accuracy                           0.93        80
   macro avg       0.93      0.93      0.92        80
weighted avg       0.93      0.93      0.92        80
```

In [14]:
```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

precision = round(precision_score(y_test, y_pred_logreg, average='micro'), 2)
recall = round(recall_score(y_test, y_pred_logreg, average='micro'), 2)
accuracy = round(accuracy_score(y_test, y_pred_logreg), 2)
f1 = round(f1_score(y_test, y_pred_logreg, average='micro'), 2)

precision,recall,accuracy,f1
```

Out[14]:
```
(0.92, 0.92, 0.92, 0.92)
```

Wall Time: 38.3s

Model dan Parameter

BERT

```
score = bert_model.evaluate(test_encoded)

print("Test Accuracy:", score[1])
```

```
2/2 [==============================] - 0s 69ms/step - loss: 0.1782 - accuracy: 0.9750
Test Accuracy: 0.9750000238418579
```

```
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score

precision = round(precision_score(y_true, y_pred, average='micro'), 2)
recall = round(recall_score(y_true, y_pred, average='micro'), 2)
accuracy = round(accuracy_score(y_true, y_pred), 2)
f1 = round(f1_score(y_true, y_pred, average='micro'), 2)

precision,recall,accuracy,f1
```

```
(0.98, 0.98, 0.98, 0.98)
```

```
[43] confusion_matrix(y_true, y_pred)
```

```
array([[24,  0],
       [ 1, 15]])
```

```
[44] print(classification_report(y_true, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.96      1.00      0.98        24
           1       1.00      0.94      0.97        16

    accuracy                           0.97        40
   macro avg       0.98      0.97      0.97        40
weighted avg       0.98      0.97      0.97        40
```

Wall Time: 1min 10s

Kesimpulan

- **BERT lebih baik dari segi akurasi**
- **Walaupun begitu, jika mengutamakan kecepatan, Machine Learning bukan pilihan yang buruk**
- **BERT mungkin lebih unggul dari sisi konsistensi, karena metode yang digunakan juga lebih rumit**

Terima Kasih ☺

Ujian Praktik – Orbit Future Academy