

Dokumentasi Sentiment Analysis

1. Library yang digunakan
 - a. Library pandas, numpy

Link Pandas : <https://pandas.pydata.org/>

Link Numpy : <https://numpy.org/>

Fungsi: Kedua library tersebut digunakan untuk manajemen dataset. Pandas digunakan untuk menyimpan dataset ke dalam suatu variabel. Numpy digunakan untuk melakukan validasi – validasi pada dataset.

- b. Library NLTK (import stopwords)

Link NLTK : <https://www.nltk.org/api/nltk.corpus.html#module-nltk.corpus>

Fungsi : Library ini digunakan untuk membuang kata – kata yang tidak berguna untuk program. Seperti kata – kata yang dikatakan terus menerus secara umum yaitu : “I”, “you”, “they”, “isn’t”, “because”, “again”, dll

Stopwords dataset “English”: <http://snowball.tartarus.org/algorithms/english/stop.txt>

- c. Library Textblob (import word)

Link Textblob : <https://textblob.readthedocs.io/en/dev/quickstart.html>

Fungsi : Library Textblob digunakan untuk lemmatize kata – kata bahasa inggris yang “Typo”, “Past Tense”, dll. Kata tersebut dijadikan sifat baku / “Present Tense” dan digunakan untuk training. Tujuan lemmatization agar tidak terlalu banyak jenis kata yang berbeda – beda Contoh : “Bought” menjadi “Buy”

- d. Library re (Regular Expression) python

Fungsi : Regular expression digunakan untuk preprocessing pada dataset training untuk digunakan.

- e. Library sklearn (Preprocessing)

Fungsi : Library ini digunakan untuk memberikan label – label kepada dataset tersebut

- f. Library sklearn.model.selection (train_test_split)

Fungsi : Library ini digunakan untuk memisahkan data – data dari dataset tersebut menjadi 2 bagian training dan testing.

g. Library `sklearn.extraction.text` (`CountVectorizer`)

Fungsi : Library ini digunakan untuk mencari fitur – fitur dari sebuah matriks text. Lalu fitur – fitur tersebut digunakan untuk membuat array matriks yang menyimpan nilai – nilai numerik mengenai umumnya kata – kata dari label tersebut diucapkan.

h. Library `sklearn.linear.model` (`SGDClassifier`)

Fungsi : Library ini digunakan untuk training model “Linear Support Vector Machine”. Hasil dari training model tersebut dapat digunakan untuk validasi dataset dari youtube subtitle.

2. Pembahasan algoritma yang dipakai

```
import pandas as pd
import numpy as np
from nltk.corpus import stopwords
from textblob import Word
import re
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score
from sklearn.linear_model import SGDClassifier
```

Import library – library yang akan digunakan.

```
# Dropping rows with other emotion labels
data = data.drop(data[data.sentiment == 'anger'].index)
data = data.drop(data[data.sentiment == 'boredom'].index)
data = data.drop(data[data.sentiment == 'enthusiasm'].index)
data = data.drop(data[data.sentiment == 'empty'].index)
data = data.drop(data[data.sentiment == 'fun'].index)
data = data.drop(data[data.sentiment == 'relief'].index)
data = data.drop(data[data.sentiment == 'surprise'].index)
data = data.drop(data[data.sentiment == 'love'].index)
data = data.drop(data[data.sentiment == 'hate'].index)
data = data.drop(data[data.sentiment == 'neutral'].index)
data = data.drop(data[data.sentiment == 'worry'].index)
```

Drop data – data dari column emotion agar model tidak training terlalu lama dan meningkatkan accuracy untuk emosi yang digunakan untuk validasi. Kita hanya menyimpan emosi “Sadness” dan “Happiness”.

```
# Making all letters lowercase
data['content'] = data['content'].apply(lambda x: " ".join(x.lower() for x in x.split()))

# Removing Punctuation, Symbols
data['content'] = data['content'].str.replace('[^\w\s]', ' ')

# Removing Stop Words using NLTK
stop = stopwords.words('english')
data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))

# Lemmatisation
data['content'] = data['content'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()])))
```

Memproses seluruh kalimat – kalimat dari dataset dengan cara, mengubah seluruh kata menjadi huruf kecil, membuang seluruh simbol – simbol yang ada, membuang kata – kata repetisi yang tidak mempunyai makna penting, dan lemmatize kata – kata.

```
#Correcting Letter Repetitions
def de_repeat(text):
    pattern = re.compile(r"(\1{2,})")
    return pattern.sub(r"\1\1", text)

data['content'] = data['content'].apply(lambda x: " ".join(de_repeat(x) for x in x.split()))

# Code to find the top 10,000 rarest words appearing in the data
freq = pd.Series(' '.join(data['content']).split()).value_counts()[-10000:]

# Removing all those rarely appearing words from the data
freq = list(freq.index)
data['content'] = data['content'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))
```

Membenarkan kata – kata yang terulang dan membuang kata – kata yang jarang digunakan dari dataset.

```
# Encoding output labels 'sadness' as '1' & 'happiness' as '0'
lbl_enc = preprocessing.LabelEncoder()
y = lbl_enc.fit_transform(data.sentiment.values)
```

Memberikan label kepada kedua fitur “Sadness” dan “Happiness”

```
# Extracting Count Vectors Parameters
count_vect = CountVectorizer(analyzer='word')
count_vect.fit(data['content'])
X_train_count = count_vect.transform(X_train)
X_val_count = count_vect.transform(X_val)

# Model : Linear SVM
lsvm = SGDClassifier(alpha=0.001, random_state=5, max_iter=15, tol=None)
lsvm.fit(X_train_count, y_train)
y_pred = lsvm.predict(X_val_count)
```

Mengubah seluruh nilai x untuk training dan validasi menjadi matriks numerik yang menyimpan umumnya kata itu diucapkan.

Menggunakan model Linear SVM untuk training dan mencari prediksi data.

```
# Find the Accuracy of the Training Model
Temp = accuracy_score(y_pred, y_val)
Accuracy = str(Temp)
print('LSVM with Count Vectors accuracy result : ' + Accuracy)
```

Mencari accuracy dari model yang sudah di training

```
# Read the text file and separate each sentence with ";" symbol
subtitle = pd.read_csv('How-Teslas-Upgrade-Over-Time.txt', sep = ";", header = None)

# Transpose the matrix from column to row and row to column
subtitle = subtitle.T

# Drop all "NULL" data from the dataset
subtitles = subtitle.dropna()
```

Membaca text file youtube subtitle (isi dari text file tersebut dapat diubah) dengan memisahkan “;” dari text filenya. Lalu menyimpan kalimat – kalimat youtube subtitle.

Transpose dataset dari “subtitle” dari column menjadi row.

Membuang seluruh “NULL” yang ada pada dataset.

```
# Doing some preprocessing on the subtitles
subtitles[0] = subtitles[0].str.replace('[^\w\s]', ' ')
stop = stopwords.words('english')
subtitles[0] = subtitles[0].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
subtitles[0] = subtitles[0].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.split()])))
```

Melakukan PreProcessing seperti diatas yaitu membuang seluruh simbol – simbol yang ada, membuang kata – kata repetisi yang tidak mempunyai makna penting, dan lemmatize kata – kata.

```
# Extracting Count Vectors feature from the subtitles
subtitle_count = count_vect.transform(subtitles[0])

# Predicting the emotion of the subtitle using trained linear SVM
subtitle_pred = lsvm.predict(subtitle_count)
print(subtitle_pred)
```

Mengambil Count Vector seperti training Model

Mencari prediksi hasil menggunakan LSVM

```
# Count all the happiness and sadness number from "subtitle_pred"
happinessCount = np.count_nonzero(subtitle_pred == 0)
sadnessCount = np.count_nonzero(subtitle_pred == 1)

# Get the percentage for both happiness and sadness
happinessPercent = happinessCount / (happinessCount + sadnessCount) * 100
sadnessPercent = sadnessCount / (happinessCount + sadnessCount) * 100

# Make the percentage into a string for printing
hpPercentage = str(happinessPercent)
sdPercentage = str(sadnessPercent)

# Print both happiness and sadness results
print("Happiness Percent = " + hpPercentage + "%")
print("Sadness Percent = " + sdPercentage + "%")
```

Menghitung total “Happiness” dan “Sadness” yang ada. Lalu mencari persentase dari kedua fitur yang ada.

Dokumentasi Youtube Subtitle Dataset

1. Library yang digunakan

a. Library pyTube3

Link : <https://python-pytube.readthedocs.io/en/latest/>

Fungsi : Menggunakan fungsi 'Youtube' yang dapat mengambil atribut dari video-video youtube melalui linknya, seperti subtitles, channel, likes, comment, dll. Khusus pada project, ini kami gunakan untuk mengekstrak subtitle yang akan diproses menjadi dataset.

b. Library Requests

Link : <https://requests.readthedocs.io/en/master/>

Fungsi : Library ini digunakan untuk mendapatkan link-link untuk video youtube yang akan diproses oleh fungsi 'Youtube' dari pyTube3.

c. Library re (Regular Expressions) python

Fungsi : Regular expression digunakan untuk preprocessing pada dataset subtitle yang didapatkan. Pre-processing ini mencakup parsing file .srt menjadi .txt.

2. Pembahasan algoritma yang dipakai

```
from pytube import YouTube
import requests
import re
```

Import library-library yang akan digunakan.

```
username = "marquesbrownlee"
url = "https://www.youtube.com/user/" + username + "/videos"
page = requests.get(url).content
data = str(page).split(' ')
item = 'href="/watch?'
vids = [line.replace('href=', 'youtube.com') for line in data if item in line]
```

Username adalah name dari channel youtube yang akan digunakan sebagai dataset. Kami menggunakan link dari youtube channel yang sudah ditentukan, lalu mengambil link dari video-video terbaru yang ada pada youtube channel tersebut. Lalu, link-link tersebut disimpan dalam sebuah list.

```
for i in range(6):
    print(vids[i]) # index the latest video
    source = YouTube(vids[i])
    en_caption = source.captions.get_by_language_code('en')
```




Dengan link-link video yang sudah disetor dalam sebuah list, kami menggunakan fungsi 'Youtube' untuk mengekstrak subtitle English yang ada pada video masing-masing.

```
en_caption_convert_to_srt =(en_caption.generate_srt_captions())








filename = source.title
filename = urlify(filename)
filename = filename + ".txt"
```

Subtitle yang diekstrak lalu akan diconvert ke bentuk .srt, diparse, dan akan dicetak ke dalam file .txt. File .txt merupakan dataset yang akan diproses oleh algoritma NLP selanjutnya.

Evaluasi

 Sentiment Analysis	6/14/2020 1:59 PM	PY File	5 KB
 text_emotion	5/25/2020 1:11 PM	Microsoft Excel C...	4,292 KB
 Youtube	6/14/2020 2:01 PM	PY File	2 KB

Sebelum Youtube code di-run file hanya terdiri dari dataset untuk training dan code sentiment analysis

 CyberTruck-Phone-Impressions-Ridiculous	6/14/2020 2:03 PM	Text Document	8 KB
 Dope-Tech-The-3500-Bluetooth-Speaker	6/14/2020 2:03 PM	Text Document	9 KB
 Reflecting-on-the-Color-of-My-Skin	6/14/2020 2:03 PM	Text Document	11 KB
 Sentiment Analysis	6/14/2020 1:59 PM	PY File	5 KB
 text_emotion	5/25/2020 1:11 PM	Microsoft Excel C...	4,292 KB
 Youtube	6/14/2020 2:03 PM	PY File	2 KB
 YouTube	6/14/2020 2:03 PM	Text Document	8 KB

Setelah di-run menghasilkan dataset yang digunakan untuk sentiment analysis

```

LSVM with Count Vectors accuracy result : 0.7832369942196532
[0 0 0 1 0 1 1 0 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 1 0 1 0
 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 1 0 1 0 0
 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0
 0 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 1 1 1 0 0
 1]
Happiness Percent = 38.17204301075269%
Sadness Percent = 61.82795698924731%

```

Setelah program sentiment analysis dijalankan menghasilkan persentase akurasi model (cukup tinggi karena kami drop label – label selain “Happiness” dan “Sadness”).

Menghasilkan matriks mengenai happiness dan sadness yang dirasakan creator video di setiap sentence dari subtitle. (“Happiness” = 0 & “Sadness” = 1)

Pada akhirnya menghasilkan persentase dari kedua “Happiness” dan “Sadness” yang dirasakan creator video. (Dapat digunakan untuk menyimpulkan perasaan yang dirasakan pada creator)

References :

Sentiment Analysis Base API - <https://github.com/aditya-xq/Text-Emotion-Detection-Using-NLP/blob/master/main.py>

Dataset untuk training Model - <https://data.world/crowdfunder/sentiment-analysis-in-text>