

LAPORAN AKHIR PRAKTIKUM

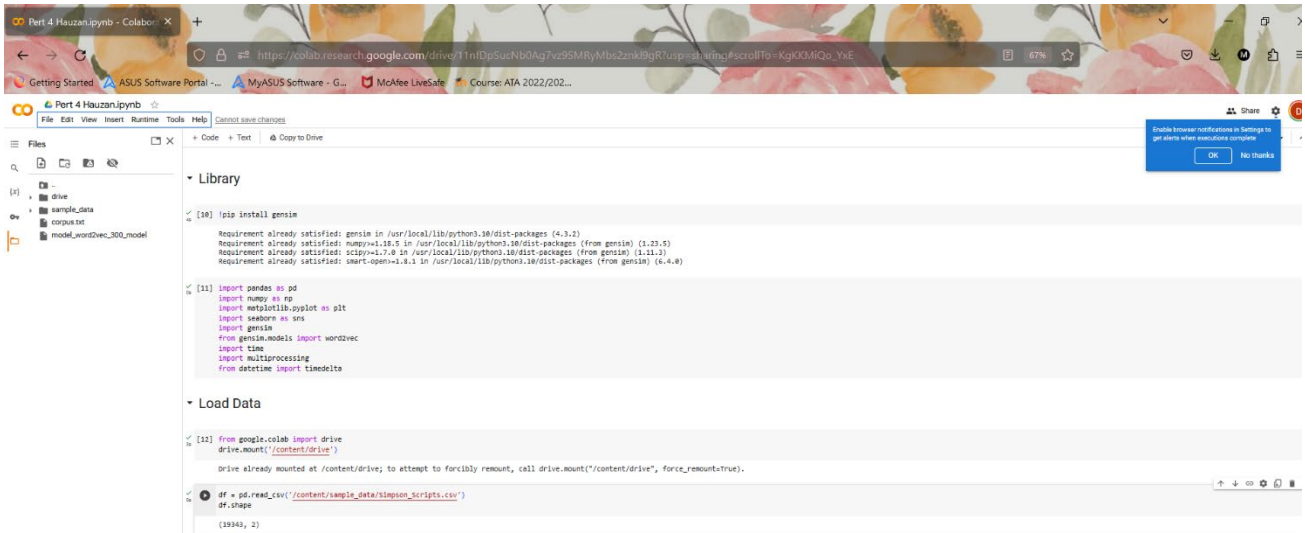
Mata Praktikum : KA
Kelas : 3IA19
Praktikum : 4
Tanggal : 11/09/2023
Materi : Natural Language Processing
NPM : 50421859
Nama : Muhamad Ariel Dwi P
Ketua Asisten :
Nama Asisten : MUHAMMAD HAUZAN DINI FAKHRI
Paraf Asisten :
Jumlah Lembar : 7



**LABORATORIUM INFORMATIKA
UNIVERSITAS GUNADARMA
2023**

LISTING

1. Screenshot Hasil kodingannya?



The screenshot shows a Jupyter Notebook titled 'Port 4 Hauzan.ipynb' in a Google Colab environment. The interface includes a file browser on the left, a code editor, and a runtime output area. The code executed includes installing gensim, importing various libraries (pandas, numpy, matplotlib, seaborn, gensim, time, multiprocessing, datetime, timedelta), and loading data from a Google Drive mount. The output shows the successful installation of gensim and the loading of a CSV file named 'Simpson_Scripts.csv' with a shape of (19343, 2).

```
[10] !pip install gensim

Requirement already satisfied: gensim in /usr/local/lib/python3.10/dist-packages (4.3.2)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.23.5)
Requirement already satisfied: scipy>=1.7.0 in /usr/local/lib/python3.10/dist-packages (from gensim) (1.11.3)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.10/dist-packages (from gensim) (6.4.0)

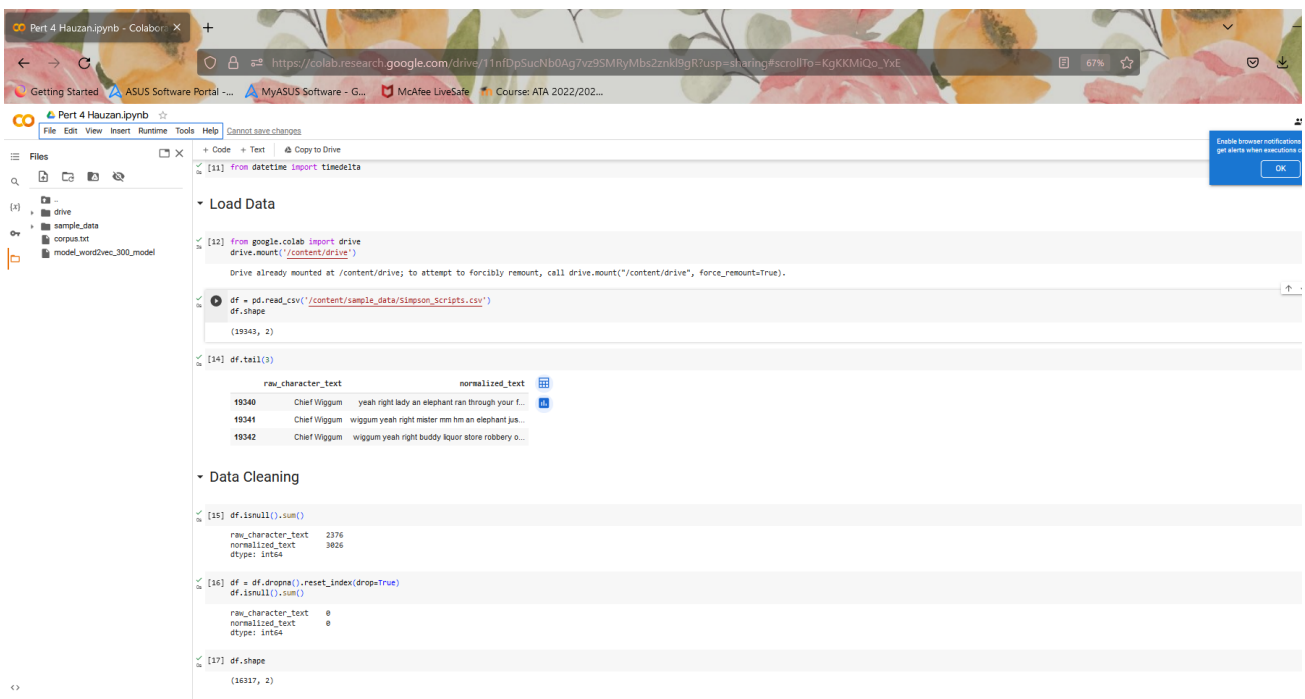
[11] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import gensim
from gensim.models import word2vec
import time
import multiprocessing
from datetime import timedelta

[12] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[13] df = pd.read_csv('/content/sample_data/Simpson_Scripts.csv')
df.shape

(19343, 2)
```



The screenshot shows the continuation of the Jupyter Notebook. The code executed includes importing datetime and timedelta, mounting Google Drive, reading the CSV file, and performing data cleaning steps. The output shows the shape of the DataFrame after dropping rows with missing values, resulting in a shape of (16317, 2). A preview of the data is also shown, displaying columns 'raw_character_text' and 'normalized_text'.

```
[11] from datetime import timedelta

[12] from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[13] df = pd.read_csv('/content/sample_data/Simpson_Scripts.csv')
df.shape

(19343, 2)

[14] df.tail(3)

raw_character_text      normalized_text
19340  Chief Wiggum      yeah right lady an elephant ran through your f...
19341  Chief Wiggum      wiggum yeah right mister mmm hm an elephant jus...
19342  Chief Wiggum      wiggum yeah right buddy liquor store robbery o...

[15] df.isnull().sum()

raw_character_text      2376
normalized_text         3026
dtype: int64

[16] df = df.dropna().reset_index(drop=True)
df.isnull().sum()

raw_character_text      0
normalized_text         0
dtype: int64

[17] df.shape

(16317, 2)
```



```

[14]: db_model.wd.most_similar(positive='merge')

[["beer", 0.898964761314582],
 ["rum", 0.898939310817611],
 ["whisky", 0.89891486797777],
 ["wine", 0.898794461502122],
 ["facebook", 0.89877098728822],
 ["beer", 0.89874818138777],
 ["us", 0.8986488118778693],
 ["beer", 0.898617883338887],
 ["united", 0.898513127326464],
 ["underground", 0.898322295599714]]

[15]: db_model.wd.similarity('reggie', 'baby')

0.8992179

[16]: db_model.wd.most_similar(positive='uconn', 'homer', negative='merge', topn=5)

[["halla", 0.89797338583338],
 ["name", 0.89789324231358],
 ["uconn", 0.89822434976251]]

[17]: db_model.wd.most_similar(positive='uconn', 'king', negative='homer')

[["for", 0.89822882881124],
 ["from", 0.89821128989894],
 ["with", 0.89870184141794],
 ["up", 0.89846798794811],
 ["new", 0.89832882881124],
 ["the", 0.89848112877777],
 ["not", 0.89847822222222],
 ["in", 0.89832882881124],
 ["is", 0.89832882881124],
 ["to", 0.89832882881124]]

[18]: db_model.wd.docvecs.get('uconn on the bus driver' + str(i))

None

```

Penjelasan Tentang Kode diatas

LIBRARY

!pip install gensim

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import gensim

from gensim.models import word2vec

import time

import multiprocessing

from datetime import timedelta

Kode yang Anda berikan melakukan beberapa hal:

!pip install gensim: Ini adalah perintah untuk menginstal library gensim menggunakan pip, yang merupakan sistem manajemen paket yang digunakan untuk menginstal dan mengelola paket perangkat lunak yang ditulis dalam Python.

import pandas as pd dan sebagainya: Ini adalah perintah untuk mengimpor beberapa library yang akan digunakan dalam kode, seperti pandas, numpy, matplotlib, seaborn, dan gensim.

from gensim.models import word2vec: Ini adalah perintah untuk mengimpor modul word2vec dari library gensim.

import time, import multiprocessing, dan from datetime import timedelta: Ini adalah perintah untuk mengimpor beberapa modul yang akan digunakan dalam kode, seperti time, multiprocessing, dan timedelta.

Load Data

```
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('/content/sample_data/Simpson_Scripts.csv')
df.shape
df.tail(3)
```

Kode yang Anda berikan melakukan beberapa hal:

`from google.colab import drive` dan `drive.mount('/content/drive')`: Ini adalah perintah untuk mengimpor modul `drive` dari library `google.colab` dan memasang Google Drive ke lingkungan Colab.

`df = pd.read_csv('/content/sample_data/Simpson_Scripts.csv')`: Ini adalah perintah untuk membaca file CSV dari Google Drive dan menyimpannya ke dalam DataFrame pandas.

`df.shape` dan `df.tail(3)`: Ini adalah perintah untuk menampilkan jumlah baris dan kolom dalam DataFrame (`df.shape`), dan menampilkan tiga baris terakhir dari DataFrame (`df.tail(3)`).

***Data Cleaning ***

```
df.isnull().sum()
df = df.dropna().reset_index(drop=True)
df.isnull().sum()
df.shape
```

Kode yang Anda berikan melakukan beberapa hal:

`df.isnull().sum()`: Ini adalah perintah untuk menghitung jumlah nilai null dalam DataFrame.

`df = df.dropna().reset_index(drop=True)`: Ini adalah perintah untuk menghapus baris yang memiliki nilai null dan mereset indeks DataFrame.

***Membuat corpus ***

```
# Menyimpan corpus ke dalam file 'corpus.txt'
corpus_path = 'corpus.txt'
with open(corpus_path, 'w') as f:
    f.write(corpus_text)
```

`corpus_text = "\n".join(df['normalized_text'])`: Ini adalah perintah untuk menggabungkan semua teks dalam kolom 'normalized_text' dari DataFrame menjadi satu string besar, dengan setiap teks dipisahkan oleh baris baru.

Training Model

```
start_time = time.time()
print('Training Word2Vec Model...')
sentences = word2vec.LineSentence(corpus_path)
w2v_model = word2vec.Word2Vec(sentences, vector_size=300,
workers=multiprocessing.cpu_count())
w2v_model.save('model_word2vec_300_model')
finish_time = time.time()
```

`sentences = word2vec.LineSentence(corpus_path)`: Ini adalah perintah untuk memuat kalimat dari file teks untuk digunakan dalam pelatihan model Word2Vec.

`w2v_model = word2vec.Word2Vec(sentences, vector_size=300, workers=multiprocessing.cpu_count())`: Ini adalah perintah untuk melatih model Word2Vec pada kalimat yang dimuat sebelumnya, dengan ukuran vektor 300 dan jumlah pekerja sebanyak jumlah core CPU.w2

*Test *

```
w2v_model.wv.similarity('woman','man')
word = 'amazing'
ms = w2v_model.wv.most_similar(word)
ms
```

`w2v_model.wv.similarity('woman','man')`: Ini adalah perintah untuk menghitung kesamaan kosinus antara vektor kata 'woman' dan 'man' dalam model Word2Vec.

Visualisasi Similarity dari data diatas

```
words, scores = zip(*ms)
```

```
# Ubah skor menjadi numpy array
```

```
scores = np.array(scores)
```

```
# Buat matriks heatmap
```

```
heatmap_data = scores.reshape(1, -1)
```

```
# Buat gambar heatmap
```

```
plt.figure(figsize=(8, 4))
```

```
plt.imshow(heatmap_data, cmap='viridis', aspect='auto')
```

```
plt.colorbar()
```

```
plt.xticks(np.arange(len(words)), words, rotation=45)
```

```
plt.yticks([]) # Menghilangkan label pada sumbu y
```

```
plt.title('Heatmap Kemiripan')
plt.show()
w2v_model.wv.most_similar(positive=['homer'])
w2v_model.wv.most_similar(positive=['marge'])
w2v_model.wv.similarity('maggie', 'baby')
w2v_model.wv.most_similar(positive=['woman', 'homer'], negative=['marge'], topn=3)
w2v_model.wv.most_similar(positive=['woman', 'king'], negative=['homer'])
w2v_model.wv.doesnt_match('walking on the bus drowned'.split())
```

`ms = w2v_model.wv.most_similar(word)`: Ini adalah perintah untuk mencari kata-kata yang paling mirip dengan kata tertentu dalam model Word2Vec.

`w2v_model.wv.most_similar(positive=['woman', 'homer'], negative=['marge'], topn=3)`: Ini adalah perintah untuk mencari kata-kata yang paling mirip dengan ‘woman’ dan ‘homer’, tetapi tidak mirip dengan ‘marge’ dalam model Word2Vec.

`w2v_model.wv.doesnt_match('walking on the bus drowned'.split())`: Ini adalah perintah untuk mencari kata yang paling tidak cocok dalam daftar kata ‘walking’, ‘on’, ‘the’, ‘bus’, dan ‘drowned’ dalam model Word2Vec.