МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“Київський політехнічний інститут ім. Ігоря Сікорського”
ФАКУЛЬТЕТ Інформатики та Обчислювальної Техніки
КАФЕДРА Інформаційних систем та технологій

Звіт до лабораторної роботи №3

з предмету: Обробка та Аналіз текстових даних на мові Python

Перевірила:
Тимофєєва Ю.С.

Виконли:
студенти групи ІК-01
Філоненко І. Р.
Гацан С. Ю.

КИЇВ - 2023

Тема: Моделі текстових даних

Мета: Ознайомитись з основними текстовими моделями та їх створення за допомогою scikit-learn та gensim.

Варіант: 11

Завдання:

Зчитати файл doc11. Вважати кожен рядок окремим документом корпусу. Виконати попередню обробку корпусу.

1. Представити корпус як модель “Сумка слів”. Вивести вектор для слова MARINER.
2. Представити корпус як модель TD-IDF. Спробувати кластеризувати документи за лопомогою ієрархічної агломераційної кластеризації.
3. Представити корпус як модель Word2Vec. Знайти подібні слова до слів mobile, athens.

Код програми

```
#!/usr/bin/env python3

import re
import numpy as np
import pandas as pd

from matplotlib import pyplot as plt

from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.metrics.pairwise import cosine_similarity
from scipy.cluster.hierarchy import dendrogram, linkage, fcluster

import nltk
from nltk.tokenize import WordPunctTokenizer
from nltk.corpus import stopwords
from gensim.models import word2vec

nltk.data.path.append("../Lab2/nltk_data")

file = open("doc11.txt", 'r').read().split('\n')
wpt = nltk.WordPunctTokenizer()

# Preparing corpus
corpus = [sentence for sentence in file if len(sentence) != 0]


def preprocess_sentence(sentence):
    sentence = re.sub(r'[^a-zA-Z\s]', '', sentence, re.I | re.A)
    sentence = sentence.lower()
    sentence = sentence.strip()
    tokens = wpt.tokenize(sentence)
    filtered_tokens = [token for token in tokens if token not in
stopwords.words('english')]
    sentence = ' '.join(filtered_tokens)
```

```
    return sentence


prepared_corpus = []

for sentence in corpus:
    prepared_corpus.append(preprocess_sentence(sentence))

print("Підготовлений корпус: ")
print(prepared_corpus)

# Task 1
cv = CountVectorizer(min_df=0., max_df=1.)
cv_matrix = cv.fit_transform(prepared_corpus)
vocab = cv.get_feature_names_out()

array = pd.DataFrame(cv_matrix.toarray(), columns=vocab)

print(array['mariner'])
# --------
# Task 2
# Отримуємо TD-IDF матрицю
tt = TfidfTransformer(norm='l2', use_idf=True)
tt_matrix = tt.fit_transform(cv_matrix)

array2 = pd.DataFrame(tt_matrix.toarray(), columns=vocab)
print("TD-IDF: ")
print(array2)
# Отримуємо матрицю подібності
similarity_matrix = cosine_similarity(tt_matrix)

array3 = pd.DataFrame(similarity_matrix)
print("\n")
print("Similarity matrix: ")
print(array3)
print('\n')

# Creating links
links = linkage(similarity_matrix, 'ward')

# Creating plot
print("Dendrogram: ")
plt.figure(figsize=(8, 3))
plt.title('Dendrogram')
plt.xlabel('Documents')
plt.ylabel('Length')
dendrogram(links)
plt.show()

# From dendrogram we can choose max distance = 1.5
max_dist = 1.6

cluster_labels = fcluster(links, max_dist, criterion='distance')

array4 = pd.DataFrame(prepared_corpus, columns=["Sentences"])
array5 = pd.DataFrame(cluster_labels, columns=["Cluster Labels ID"])
array6 = pd.concat([array4, array5], axis=1)
print(array6)
print("\n\n")
# --------
# Task 3
tokenized_prepared_corpus = [wpt.tokenize(sentence) for sentence in
prepared_corpus]
vector_size = 100
```

```
window = 30
min_count = 1
sample = 1e-3
w2v_model = word2vec.Word2Vec(tokenized_prepared_corpus,
                              vector_size=vector_size,
                              window=window,
                              min_count=min_count,
                              sample=sample)
# Similar words
try:
    for search_term in ["mobile", "athens"]:
        print({search_term: [
                             item[0]
                             for item
                             in w2v_model.wv.most_similar(
                                                          [search_term],
                                                          topn=5
                                                         )
                             ]})
except KeyError as e:
    print(e)
# --------
```

Результат роботи програми

Підготовлений корпус:

['mariner sent mars expanded upon work done mariner four years earlier', 'back spotlight public scrutiny attempt erase double disappointment athens olympics', 'mariner ended atlantic ocean rocket launcher autopilot failed', 'million people mobile uk mobile operators keen encourage people move onto sophisticated handsets', 'mariner sister probe mariner reach mars', 'recent survey said mobile owners thinking upgrading g p hones', 'radcliffe concedes probably learn lot bad experiences athens time']

0 2
1 0
2 1
3 0
4 2
5 0
6 0
Name: mariner, dtype: int64

TD-IDF:

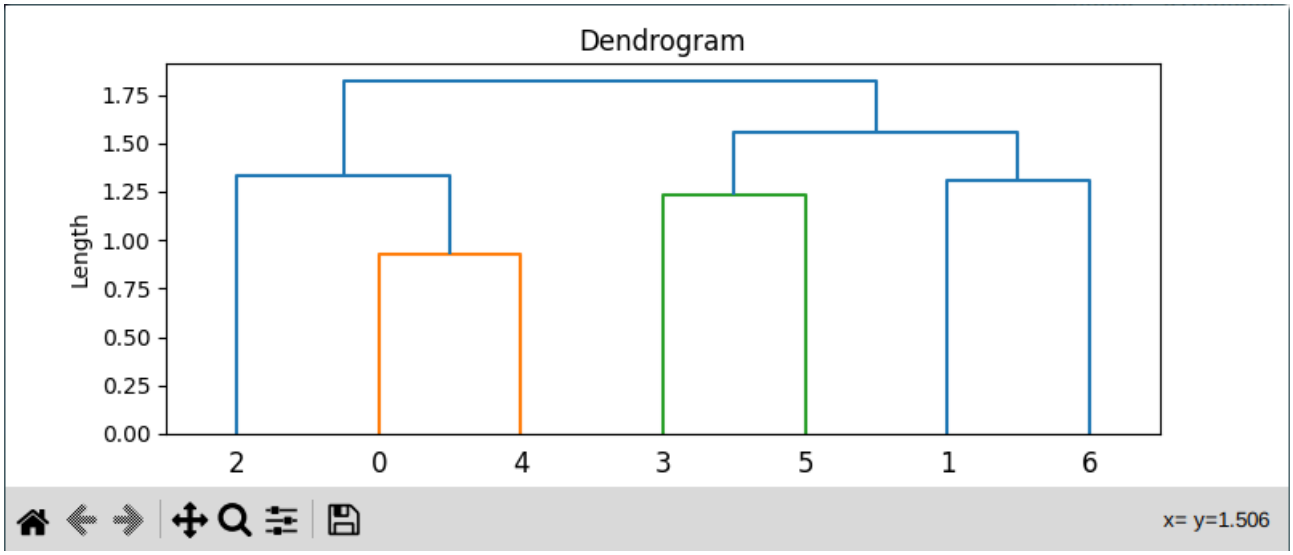
	athens	atlantic	attempt	autopilot	back	bad	...	time	uk	upgrading	upon	work	years
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.305669	0.305669	0.305669
1	0.266675	0.000000	0.321262	0.000000	0.321262	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.365065	0.000000	0.365065	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.251927	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.360632	0.000000	0.000000	0.000000
6	0.281603	0.000000	0.000000	0.000000	0.000000	0.339245	...	0.339245	0.000000	0.000000	0.000000	0.000000	0.000000

[7 rows x 56 columns]

Similarity matrix:

	0	1	2	3	4	5	6
0	1.000000	0.000000	0.112355	0.000000	0.345954	0.000000	0.000000
1	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.075096
2	0.112355	0.000000	1.000000	0.000000	0.153921	0.000000	0.000000
3	0.000000	0.000000	0.000000	1.000000	0.000000	0.125203	0.000000
4	0.345954	0.000000	0.153921	0.000000	1.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.125203	0.000000	1.000000	0.000000
6	0.000000	0.075096	0.000000	0.000000	0.000000	0.000000	1.000000

Dendrogram:



	Sentences	Cluster Labels	ID
0	mariner sent mars expanded upon work done mari...		1
1	back spotlight public scrutiny attempt erase d...		2
2	mariner ended atlantic ocean rocket launcher a...		1
3	million people mobile uk mobile operators keen...		2
4	mariner sister probe mariner reach mars		1
5	recent survey said mobile owners thinking upgr...		2
6	radcliffe concedes probably learn lot bad expe...		2

```
{'mobile': ['expanded', 'years', 'keen', 'double', 'public']}
{'athens': ['double', 'g', 'million', 'time', 'probably']}
```