

Національний технічний університет України «КПІ ім. Ігоря Сікорського»
Факультет Інформатики та Обчислювальної Техніки
Кафедра інформаційних систем та технологій

Лабораторна робота № 3

з дисципліни «Обробка та аналіз текстових даних на Python»

На тему:

«Моделі текстових даних»

Варіант №3

Виконала:

студентка групи ІС-12.

Мельникова К.О.

Перевірила:

Тимофєєва Ю. С.

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

Завдання до лабораторної роботи

Зчитати файл doc3. Вважати кожен рядок окремим документом корпусу.

Виконати попередню обробку корпусу.

Лабораторна робота №3

```
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.cluster import AgglomerativeClustering
from gensim.models import FastText

# Зчитуємо текст з файлу doc3.txt
with open("data.txt", "r", encoding="utf-8") as file:
    corpus = file.readlines()

# Видаляємо зайві пробіли та символи нового рядка з кожного документу
corpus = [doc.strip() for doc in corpus]
```

[5]

✓ 0.0s

Python

1)Представити корпус як модель «Сумка слів». Вивести вектор для слова Google.

1. Представлення корпусу як модель "Сумка слів"

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)

# Вивід вектора для слова "Google"
word_index = vectorizer.vocabulary_.get("google")
google_vector = X[:, word_index].toarray().reshape(-1)
print("Bag of Words вектор для слова 'Google':", google_vector)
```

[16]

✓ 0.0s

Python

... Bag of Words вектор для слова 'Google': [1 1 0 0 0 0]

2)Представити корпус як модель TF-IDF. Спробувати кластеризувати документи за допомогою ієрархічної агломераційної кластеризації.

2. Представлення корпусу як модель TF-IDF та кластеризація документів

```
tfidf_vectorizer = TfidfVectorizer()
X_tfidf = tfidf_vectorizer.fit_transform(corpus)
cluster = AgglomerativeClustering(n_clusters=2, metric='cosine', linkage='complete')
cluster.fit(X_tfidf.toarray())
```

```
# Вивід результатів кластеризації
for i in range(len(corpus)):
    print(f"Документ {i} належить до кластера {cluster.labels_[i]}")
```

[17] ✓ 0.0s

Python

```
... Документ 0 належить до кластера 0
Документ 1 належить до кластера 0
Документ 2 належить до кластера 0
Документ 3 належить до кластера 1
Документ 4 належить до кластера 1
Документ 5 належить до кластера 0
```

3) Представити корпус як модель FastText. Знайти подібні слова до слів *turkey*, *mummies*.

3. Представлення корпусу як модель FastText та знаходження схожих слів

```
tokenized_corpus = [doc.lower().split() for doc in corpus]
model = FastText(tokenized_corpus, vector_size=100, window=5, min_count=1, workers=4, sg=1)
similar_words_turkey = model.wv.most_similar("turkey")
similar_words_mummies = model.wv.most_similar("mummies")

print("Схожі слова на 'turkey':", similar_words_turkey)
print("Схожі слова на 'mummies':", similar_words_mummies)
```

[18] ✓ 0.8s

Python

```
... Схожі слова на 'turkey': [('turkey', 0.5891311168670654), ('study', 0.2828451097011566), ('aggressive', 0.2146170735359192)]
Схожі слова на 'mummies': [('mummy', 0.33721062541007996), ('physical', 0.2892851233482361), ('museum', 0.2338668555021286),
```