МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №16**

з дисципліни «Програмування мовою Python»

на тему «**Використання Python бібліотек**»

Виконав студент групи КНз-31с

Коноз Р.В.

Перевірила Парфененко Ю.В.

Варіант 3

Суми 2024

**Завдання 1**

1. **Постановка задачі**

Імпортувати бібліотеку NLTK та тексти із електронного архіву  текстів Project Gutenberg, для виконання завдань взяти текст, заданий варіантом.

austen-sense.txt

Визначити кількість слів у текст.

Визначити 10 найбільш вживаних слів у тексті, побудувати на основі цих даних стовпчасту діаграму.

Виконати видалення з тексту стоп-слів та пунктуації, після чого знову знайти 10 найбільш вживаних слів у тексті та побудувати на їх основі стовпчасту діаграму.

1. **Текст програми**

import matplotlib.pyplot as plt  
from collections import Counter  
  
  
def extract\_stopwords(text):  
 filename = "stopwords.txt"  
 stopwords\_file = open(filename, "r",  
 encoding="utf-8") # https://github.com/igorbrigadir/stopwords/blob/master/en/terrier.txt  
 stopwords = [word.strip() for word in stopwords\_file]  
 clean\_text = [word.lower().strip("\"'()“\n").rstrip("\",.'()“\n") for word in text if not word.lower() in stopwords]  
 ' '.join(clean\_text)  
 return clean\_text  
  
  
def extract\_articles(text):  
 articles=["a","the"]  
 clean\_text=' '.join([])  
  
  
def main():  
 def Open(name, mode):  
 try:  
 return open(name, mode, encoding="utf-8")  
 except:  
 print(f"There was an error while opening file {name}")  
 return None  
  
 def Read(file):  
 file.seek(0)  
 text = file.read().split()  
 return text  
  
 def count\_words(text):  
 words\_count = 0  
 for word in text:  
 if not word.isdigit():  
 words\_count += 1  
 return words\_count  
  
 def most\_used\_words(text):  
 counter = Counter(text)  
 most\_used = counter.most\_common(10)  
 x = [most\_used[el][0] for el in range(len(most\_used))]  
  
 y = [most\_used[el][1] for el in range(len(most\_used))]  
 plt.bar(x, y)  
 plt.title("10 найбільш вживаних слів у тексті")  
 plt.xlabel("Слова")  
 plt.ylabel("Зустрічаються разів у тексті")  
 plt.tight\_layout()  
 plt.show()  
 return most\_used  
  
 mypath = "austen-sense.txt"  
 file = Open(mypath, "r")  
 counter = count\_words(Read(file))  
 print("Word counter:", counter)  
 most\_used = most\_used\_words(Read(file))  
  
 clean\_text = extract\_stopwords(Read(file))  
  
 most\_used\_cleaned = most\_used\_words(clean\_text)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

1. **Приклад роботи програми**



**До видалення стоп слів**

**A graph of blue bars

Description automatically generated**

**Після видалення стоп слів**

**A graph of blue bars

Description automatically generated**

**Завдання 2**

1. **Постановка задачі**

Зберегти у текстовому файлі довільний текст до 100 слів.

Виконати наступні дії над текстом:

1. Токенізація по словам
2. Лемматизація та стеммінг
3. Видалення стоп-слів
4. Видалення пункуації

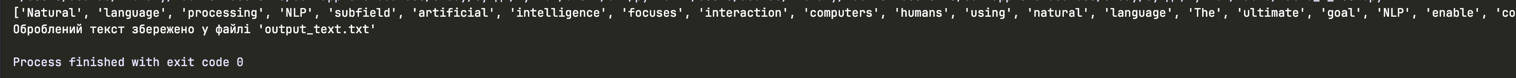
Записати оброблений текст у інший файл.

Для виконання завдань лабораторної роботи використати бібліотеку NLTK

1. **Текст програми**

*import* nltk  
*from* nltk.tokenize *import* word\_tokenize  
*from* nltk.corpus *import* stopwords  
*from* nltk.stem *import* WordNetLemmatizer, PorterStemmer  
*import* string, textwrap  
  
# Завантаження необхідних даних  
nltk.download("punkt")  
nltk.download("stopwords")  
nltk.download("wordnet")  
  
  
# Створення файлу з текстом  
*with* open("input\_text.txt", "w", encoding="utf-8") *as* file:  
 file.write("""  
Natural language processing (NLP) is a subfield of artificial intelligence that focuses   
on the interaction between computers and humans using natural language.   
The ultimate goal of NLP is to enable computers to understand, interpret, and generate human languages.  
 """)  
# Читання тексту з файлу  
*with* open("input\_text.txt", "r", encoding="utf-8") *as* file:  
 text = file.read()  
  
# Токенізація  
tokens = word\_tokenize(text)  
  
# Лемматизація  
lemmatizer = WordNetLemmatizer()  
lemmatized\_tokens = [lemmatizer.lemmatize(token) *for* token *in* tokens]  
  
# Стеммінг  
stemmer = PorterStemmer()  
stemmed\_tokens = [stemmer.stem(token) *for* token *in* lemmatized\_tokens]  
  
# Видалення стоп-слів  
stop\_words = set(stopwords.words("english"))  
filtered\_tokens = [token *for* token *in* stemmed\_tokens *if* token.lower() *not in* stop\_words]  
  
# Видалення пунктуації  
processed\_tokens = [token *for* token *in* filtered\_tokens *if* token *not in* string.punctuation]  
  
# Об'єднання результату  
processed\_text = " ".join(processed\_tokens)  
  
# Запис у новий файл  
*with* open("output\_text.txt", "w", encoding="utf-8") *as* file:  
 file.write(textwrap.fill(processed\_text, 100))  
  
print("Оброблений текст збережено у файлі 'output\_text.txt'")

1. **Приклад роботи програми**



A black screen with white text

Description automatically generated

All the work can be found on my GitHub repository: <https://github.com/Dedukr/UkrUniStuff>