МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №16**

з дисципліни «Програмування мовою Python»

на тему «Обробка природної мови з використанням Python бібліотек»

Виконав студент групи КН-33

Глумний Тимур

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.

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. Код програми

import nltk

from collections import Counter

import matplotlib.pyplot as plt

import string

# Завантаження необхідних ресурсів NLTK

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('gutenberg')

# Завантаження тексту з архіву Project Gutenberg

text = nltk.corpus.gutenberg.raw('austen-sense.txt')

# Токенізація тексту на слова

words = nltk.word\_tokenize(text)

# Підрахунок кількості слів

word\_count = len(words)

# Підрахунок частоти кожного слова

word\_freq = Counter(words)

# Отримуємо 10 найбільш вживаних слів

most\_common\_words = word\_freq.most\_common(10)

# Збереження результатів у текстовий файл

with open('word\_analysis.txt', 'w', encoding='utf-8') as file:

file.write(f"Кількість слів у тексті (до видалення): {word\_count}\n\n")

file.write("10 найбільш вживаних слів (до видалення):\n")

for word, count in most\_common\_words:

file.write(f"{word}: {count}\n")

# Створення графіка для 10 найбільш вживаних слів

words\_list = [word for word, count in most\_common\_words]

counts\_list = [count for word, count in most\_common\_words]

plt.figure(figsize=(10, 6))

plt.bar(words\_list, counts\_list, color='skyblue')

plt.title('10 найбільш вживаних слів (до видалення)')

plt.xlabel('Слова')

plt.ylabel('Кількість повторень')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('word\_frequency\_graph.png')

# Функція для видалення пунктуації, стоп-слів і артиклів

def remove\_punctuation\_stopwords\_and\_articles(words):

stop\_words = set(nltk.corpus.stopwords.words('english'))

articles = {'a', 'an', 'the'}

cleaned\_words = [

word for word in words

if word.lower() not in stop\_words and word.lower() not in articles and word not in string.punctuation

]

return cleaned\_words

# Видаляємо пунктуацію, стоп-слова та артиклі

cleaned\_words = remove\_punctuation\_stopwords\_and\_articles(words)

# Підрахунок кількості слів після видалення

cleaned\_word\_count = len(cleaned\_words)

# Підрахунок частоти кожного слова після видалення

cleaned\_word\_freq = Counter(cleaned\_words)

# Отримуємо 10 найбільш вживаних слів після видалення

most\_common\_cleaned\_words = cleaned\_word\_freq.most\_common(10)

# Збереження результатів після видалення у файл

with open('cleaned\_word\_analysis.txt', 'w', encoding='utf-8') as file:

file.write(f"Кількість слів після видалення: {cleaned\_word\_count}\n\n")

file.write("10 найбільш вживаних слів після видалення пунктуації, стоп-слів та артиклів:\n")

for word, count in most\_common\_cleaned\_words:

file.write(f"{word}: {count}\n")

# Створення графіка для 10 найбільш вживаних слів після видалення

cleaned\_words\_list = [word for word, count in most\_common\_cleaned\_words]

cleaned\_counts\_list = [count for word, count in most\_common\_cleaned\_words]

plt.figure(figsize=(10, 6))

plt.bar(cleaned\_words\_list, cleaned\_counts\_list, color='lightcoral')

plt.title('10 найбільш вживаних слів після видалення пунктуації, стоп-слів та артиклів')

plt.xlabel('Слова')

plt.ylabel('Кількість повторень')

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('cleaned\_word\_frequency\_graph.png')

print("Аналіз завершено. Результати збережено у файлах.")  
3. Результат виконання програми

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Графік, ряд

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

**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.

1. Код програми

import nltk

import string

import re

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer

from nltk.stem import PorterStemmer

# Завантажуємо необхідні ресурси NLTK

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

# Довільний текст з великим використанням лапок і пунктуації

input\_text = """

" Agriculture is an important branch of economy. Economic growth of any country depends on the development of agriculture which supplies people with food and clothing and industry with raw materials." "

The word “agre” is a Latin word. It means the cultivation of fields in order to grow crops. Now agriculture also includes the use of land to breed farm animals. ' ' ' ' ' ' ' '

We do not know when people began to grow crops. It was many thousand years ago. Now crop production and animal husbandry are highly developed branches of agriculture. .. . .. ..

Life is impossible without plants. They play a highly important role in everyday life of people. Plants that are grown by farmers are known as farm crops. They ""are used for many different purposes. Most of them are used directly "" as food for people, some are consumed by farm animals, others are used in industry and medicine. ""

Crop production is the practice of growing and harvesting crops. The most important ' ' crops grown by man are grain crops, vegetables , , , , ......and grasses. In order to obtain high yields crops are grown under favourable soil and climatic conditions.

"""

# 1. Видалення лапок, трикрапки та пунктуації перед токенізацією

input\_text\_no\_quotes = re.sub(r"['\"]+", "", input\_text) # Видаляємо всі лапки (одинарні і подвійні)

input\_text\_no\_punctuation = re.sub(r"[^\w\s]", " ", input\_text\_no\_quotes) # Видаляємо всю пунктуацію, окрім слів та пробілів

input\_text\_no\_dots = re.sub(r"\.{3,}", " ", input\_text\_no\_punctuation) # Видаляємо трикрапку (послідовність крапок)

# 2. Токенізація по словам

words = word\_tokenize(input\_text\_no\_dots)

# 3. Лемматизація та стеммінг

lemmatizer = WordNetLemmatizer()

stemmer = PorterStemmer()

lemmatized\_words = [lemmatizer.lemmatize(word) for word in words] # Лемматизація

stemmed\_words = [stemmer.stem(word) for word in words] # Стеммінг

# 4. Видалення стоп-слів

stop\_words = set(stopwords.words('english'))

filtered\_words = [word for word in lemmatized\_words if word.lower() not in stop\_words]

# 5. Видалення пунктуації (потрібно після видалення пунктуації та лапок)

final\_words = [word for word in filtered\_words if word not in string.punctuation]

# Запис обробленого тексту в новий файл

processed\_text = ' '.join(final\_words)

# Записуємо результат в файл

with open('processed\_text.txt', 'w', encoding='utf-8') as file:

file.write(processed\_text)

print("Оброблений текст збережено в файл 'processed\_text.txt'.")

1. Результат виконання проограми  
   

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис  
**https://github.com/doomedd/python-laba-16**