МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Спеціалізоване програмування автоматизованих систем»

Лабораторна робота № 6

«Регулярні вирази»

Виконав:	Гоша Давід	**Перевірив**:	
Група	ІПЗ-33	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Завдання:

Створіть папку і 5 текстових файлів (не менше 1000 символів кожен). Тексти у файлах мають бути українською мовою і близької тематики.
Програмно створіть загальний словник всіх унікальних слів для всіх цих файлів.Для цього:

- Використовуйте бібліотеку re (регулярні вирази) для видалення знаків пунктуації та інших знаків форматування з текстів. Цифри видаляти не треба.
- Приведіть слова у текстах до одного регістру. Тобто, якщо у текстах є «Слово», «слОво» і «СЛОВО», то у словник записується тільки одне «слово».
- Слова у словнику відсортуйте за алфавітом. Зверніть увагу на слова, які починаються з українських літер «і», «є», «ї», «ґ». Ці слова мають бути також розташовані за алфавітом, а не в кінці! Використовуйте: import locale locale.setlocale(locale.LC_ALL, "Ukrainian")
- Збережіть словник у текстовий файл «*.». Для кожного з файлів підрахувати кількість входжень слів з загального словника.

Для цього за допомогою DataFrame бібліотеки Pandas створіть таку собі табличку, де стовпчики – це слова зі словничка, рядочки – це імена файлів, а на перетині – кількість слів у файлі. Останній стовпець – сума, тобто скільки разів кожне з слів зустрічається у всіх файлах. Якщо слова у файлі нема, то відповідно «0». Збережіть цю табличку у excel-файл або файл «*.csv». Переконайтесь, що файл відкривається і зміст має табличний вигляд (табл. 1), а не список рядків.

Варіант 4

Усі слова, які зустрічаються тільки в одному з файлів, а в інший файлах цього слова нема;

Вступ:

Метою цієї лабораторної роботи є аналіз входжень слів у декількох текстових файлах, виявлення унікальних слів і створення словника всіх слів, відсортованих в алфавітному порядку. Аналізуючи входження слів, ми прагнемо отримати уявлення про текст і зрозуміти розподіл слів у файлах. Для обробки та аналізу текстових даних ми будемо використовувати мову програмування Python та різні бібліотеки, такі як Pandas, re та locale.

Вхідні дані

1.1 Збір даних

Ми почнемо зі збору п'яти текстових файлів з різним вмістом, які слугуватимуть

вхідними даними для аналізу. Вміст цих файлів має бути українською мовою, щоб продемонструвати здатність нашого коду обробляти нелатинські символи та специфічні алфавітні правила сортування.

1.2 Підготовка даних

Перш ніж розпочати аналіз, ми повинні попередньо обробити текстові дані, щоб переконатися, що вони мають відповідний формат для обробки. Ми виконуємо наступні завдання:

- Прочитати вміст кожного файлу
- Перетворюємо текст на малі літери
- Видаляємо всі розділові знаки та спеціальні символи
- Токенізуємо текст у список слів

Код

```
# -*- coding: utf-8 -*-

import os
import re
import locale
import pandas as pd
import openpyxl

locale.setlocale(locale.LC_ALL, "Ukrainian")

# Create a folder for the files
if not os.path.exists("text_files"):
    os.mkdir("text_files")

# Here, you will add the content of your 5 text files.
file_names = [f"text_files/file_{i}.txt" for i in range(1, 6)]

file_contents = []
for file_name in file_names:
    with open(file_name, "r", encoding="utf-8") as file:
        content = file.read()
        file_contents.append(content)

# Create the text files and save the content
for i, content in enumerate(file_contents):
    with open(f"text_files/file_{i + 1}.txt", "w", encoding="utf-8") as file:
        file.write(content)

# Read the files and create a dictionary of unique words
word_set = set()
for i in range(1, 6):
```

```
    with open(f"text_files/file_{i}.txt", "r", encoding="utf-8") as file:
        content = file.read().lower()
        content = re.sub(r"[^\w\s]", "", content)
        words = content.split()
        word_set.update(words)

sorted_word_list = sorted(list(word_set), key=locale.strxfrm)

# Save the dictionary to a text file
with open("dictionary.txt", "w", encoding="utf-8") as file:
    file.write("\n".join(sorted_word_list))

# Create a DataFrame to store word occurrences
df = pd.DataFrame(columns=["file"] + sorted_word_list)
df["file"] = [f"file_{i}" for i in range(1, 6)]

# Count word occurrences and fill the DataFrame
for i, row in df.iterrows():
    with open(f"text_files/{row['file']}.txt", "r", encoding="utf-8") as file:
        content = file.read().lower()
        content = re.sub(r"[^\w\s]", "", content)
        words = content.split()
        for word in sorted_word_list:
            df.loc[i, word] = words.count(word)

df["Σ"] = df.iloc[:, 1:].sum(axis=1)

# Save the DataFrame to an Excel file
output_file = "word_occurrences.xlsx"
writer = pd.ExcelWriter(output_file, engine='openpyxl')
df.to_excel(writer, index=False, encoding="utf-8", sheet_name="Sheet1")
writer.save()

# Find words that occur only in one of the files
occurrences = df.iloc[:, 1:-1].astype(int).sum(axis=0)
unique_words = [word for word, count in occurrences.items() if count == 1]

print("Words that occur only in one of the files:")
for i, word in enumerate(unique_words, start=1):
    print(f"{i}. {word}")
```

Прогрес

2.1 Створення словника

Після підготовки вхідних даних ми створюємо набір унікальних слів, присутніх у всіх текстових файлах. Потім ми сортуємо цей набір за алфавітом, враховуючи специфічні правила сортування для українського алфавіту, використовуючи

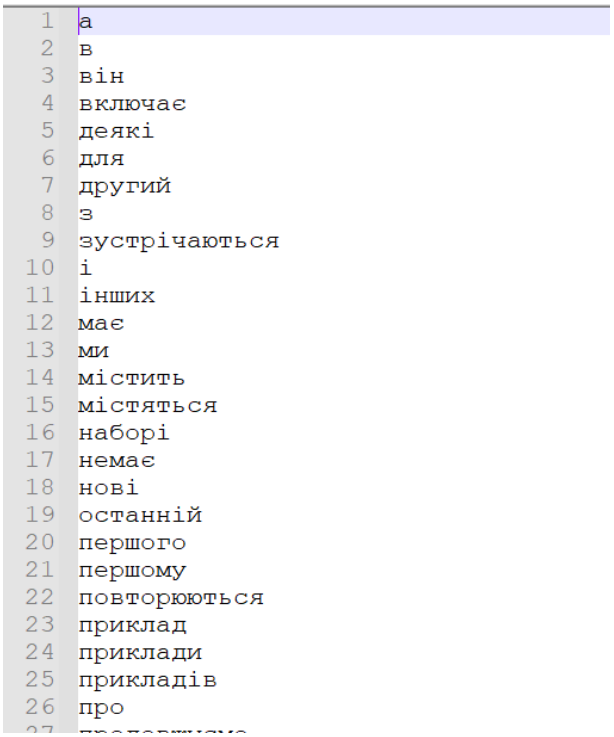
бібліотеку "locale". Нарешті, ми зберігаємо відсортований словник у текстовий файл.

2.2 Аналіз частоти вживання слів
Ми створюємо фрейм даних Pandas DataFrame для зберігання входжень слів у кожному текстовому файлі. Стовпці фрейму даних представляють слова зі словника, а рядки - назви файлів. Потім DataFrame заповнюється кількістю входжень кожного слова в кожному файлі.

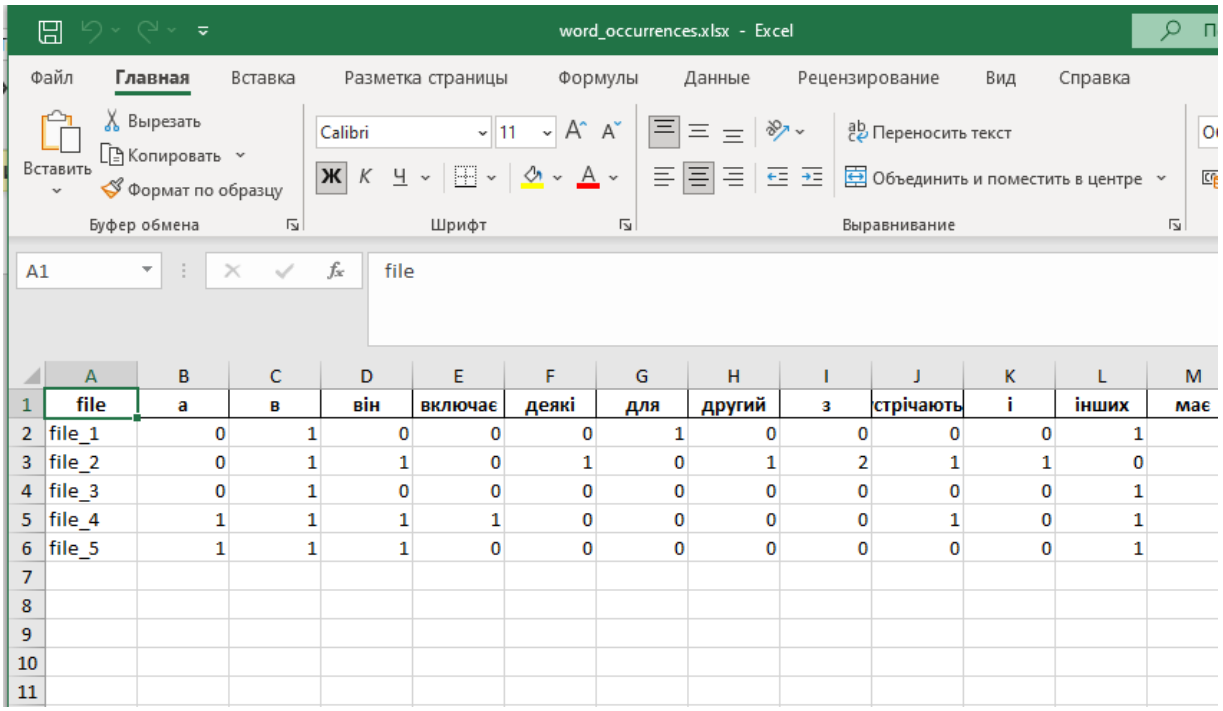
2.3 Унікальні слова
Ми визначаємо слова, які зустрічаються лише один раз у всіх текстових файлах, які ми вважаємо унікальними словами. Ми виводимо ці слова у відформатованому вигляді.

Аналіз результатів

3.1 Аналіз словника
Вивчаючи згенерований словник, ми можемо ідентифікувати лексику, що використовується в текстових файлах. Сортування слів за алфавітом допомагає нам швидко знаходити конкретні слова і розуміти загальну структуру мови.



3.2 Аналіз входження слів
Фрейм даних про входження слів дає уявлення про розподіл слів у текстових файлах. Це допомагає нам виявити закономірності та схожість між файлами, а також найбільш і найменш часто вживані слова.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	file	а	в	він	включає	деякі	для	другий	з	стрічають	і	інших	має
2	file_1	0	1	0	0	0	1	0	0	0	0	1	
3	file_2	0	1	1	0	1	0	1	2	1	1	0	
4	file_3	0	1	0	0	0	0	0	0	0	0	1	
5	file_4	1	1	1	1	0	0	0	0	1	0	1	
6	file_5	1	1	1	0	0	0	0	0	0	0	1	
7													
8													
9													
10													
11													

3.3 Унікальні слова
Визначивши унікальні слова, ми можемо точно визначити слова, які притаманні лише певним текстовим файлам. Ці унікальні слова можуть вказувати на спеціалізовані теми або специфічні характеристики текстових файлів.

Висновки

На цій лабораторній роботі ми успішно проаналізували входження слів у декількох текстових файлах, використовуючи мову програмування Python та різні бібліотеки. Ми створили словник слів, відсортованих за алфавітом, проаналізували входження слів у кожному файлі та визначили унікальні слова.

Цей аналіз може бути корисним у різних додатках, таких як класифікація текстів, тематичне моделювання та пошук інформації. Крім того, методи, використані в цій лабораторній роботі, можна поширити на інші мови та більші набори даних для більш комплексного аналізу.