НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»
Факультет Інформатики та обчислювальної техніки
Кафедра Інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА №2

з дисципліни «Обробка та аналіз текстових даних на Python»
на тему «Використання регулярних виразів для попередньої обробки тексту»
Варіант №2

Виконала: студентка групи ІС-з21
Коломієць Катерина Миколаївна
23.05.2025

Перевірив: асист. Мягкий М. Ю.

Київ – 2025

GitHub: https://github.com/kate-miiets-uni/oatd-py

Хід роботи

1. Імпортуємо бібліотеки requests та re. За допомогою функції get бібліотеки requests надсилаємо запит про отримання тексту веб-сторінки на сервер.

```
import requests
import re

url = 'https://www.bbc.com/ukrainian/articles/c249qgylz5eo'

# Надсилаємо запит на отримання тексту
response = requests.get(url)
```

2. Робимо перевірку на відповідь сервера на запит. Якщо все добре, status_code==200 і виконуємо фільтрування тексту на пунктуацію. Здійснюємо це за допомогою функції .sub() бібліотеки re і регулярного виразу r'\W+', який заміняє усі символи, окрім літер цифр та нижніх підкреслень, на пустий рядок ''.

```
    # Очищуємо текст від пунктуації
    clean_text = re.sub( pattern: r'\W+',  repl: '', text)
```

3. Далі знаходимо в тексті усі телефонні номери за допомогою функції бібліотеки re .findall() та регулярного виразу r'\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}'. Функція проходиться по усьому рядку та записує кожен номер, який підійшов по формату регулярного виразу, в окремий список.

```
    # Знаходимо телефонні номери в тексті
    phone_numbers = re.findall(  pattern: r'\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}', clean_text)
```

4. Виводимо на екран увесь очищений текст та загальну кількість знайдених телефонних номерів.

```
    # Виводимо очищений текст та кількість знайдених номерів
    print(f"Очищений текст: {clean_text}")
    print(f"Знайдено телефонних номерів :{len(phone_numbers)}")
```

5. У випадку, якщо status_code==404 або ==500, на консоль виведеться повідомлення.

```
# Якщо запит невдалий
else:
    print("Не вдалося завантажити сторінку.")
```

Результат програми:

```
Очищений текст:
 doctypehtmlhtmllangukclassnojsdirltrheadmetadatareacthelmettruecharSetutf8metadatareacthelmettruenameviewportconten
 twidthdevicewidthinitialscale1minimumscale1scriptasyncsrchttpsmybbcanalyticsfilesbbcicoukreverbclientjsreverb392jss
 criptscriptdatareacthelmettrueasynctypetextjavascriptsrchttpsstaticfilesbbcicoukwssimorghassetspublicstaticjscomsco
 remain10jsscriptlinkdatachunkmainrelasscripthrefhttpsstaticfilesbbcicoukwssimorghassetspublicstaticjsmodernframewor
 kc7b45eed9346da44jscrossoriginanonymouslinkdatachunkmainrelasscripthrefhttpsstaticfilesbbcicoukwssimorghassetspubli
 cstaticjsmodernframeworke5bca7e4df406d1ejscrossoriginanonymouslinkdatachunkmainrelasscripthrefhttpsstaticfilesbbcic
 [illegible]
 ymousnoModulescriptscriptdatachunkukrainiansrchttpsstaticfilesbbcicoukwssimorghassetspublicstaticjslegacyukrainian7
 e477069jsdefercrossoriginanonymousnoModulescriptscriptdatachunkthemesukrainiansrchttpsstaticfilesbbcicoukwssimorgha
 ssetspublicstaticjslegacythemesukrainian8dcae40djsdefercrossoriginanonymousnoModulescriptscriptdatachunkfrosted_pro
 mosrchttpsstaticfilesbbcicoukwssimorghassetspublicstaticjslegacyfrosted_promoab316804jsdefercrossoriginanonymousnoM
 odulescriptdivendifdivscripttypetextjavascriptdocumentdocumentElementclassListremovenojsscriptbodyhtml
Знайдено телефонних номерів :922
```

Висновки

Отже, у ході лабораторної роботи я завантажила текстову новину з сайту, поглибила знання у використанні модуля re, а саме дізналася про його функції .search() і .findall(), та за допомогою однієї з функцій змогла знайти в очищеному від пунктуації тексті необхідні дані, а саме телефонні номери.

Відповіді на контрольні запитання

1. Як виглядає регулярний вираз для пошуку телефонних номерів?

Регулярний вигляд для пошуку телефонних номер має вигляд r'\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}', він дозволяє розпізнати номери у форматах (XXX) XXX-XXXX, XXX-XXX-XXXX або XXX.XXX.XXXX.

2. Як видалити пунктуацію за допомогою регулярних виразів?

Видалити пунктуацію за допомогою регулярного виразу r'\W+. Метасимволи у виразі означають наступне: [] - визначають набір символів для пошуку відповідності, \W - відповідає будь-якому не буквенно-цифровому символу. r'\W+' означає шукати всі символи, які НЕ є літерами, цифрами, підкресленням.

3. Що означає символ \W у регулярних виразах?

\W — відповідає будь-якому не буквенно-цифровому символу, рівнозначне [^a-zA-Z0-9_].

4. Які функції модуля re можна використати для очищення тексту?

У модулі re можна використовувати функцію sub() – приймає три обов’язкових аргументи: pattern – регулярний вираз або рядок, за яким треба здійснювати пошук по рядку, replace – символ або рядок, на який необхідно замінити знайдену за регулярним виразом частину рядка, string – рядок, по якому необхідно зробити пошук. Також може приймати ще два необов’язкових параметри: count - максимальна кількість замін, flags - додаткові параметри.

5. Які проблеми можуть виникнути при пошуку телефонних номерів?

При пошуку телефонних номерів можна стикнутися з такими проблемами:

- різні формати номерів: +380 98 765 4321 або 098-765-4321, або (098) 765 4321, або 098.765.4321. Через при пошуку телефонних номерів варто використовувати гнучкі регулярні вирази;

- не-телефонні номери або набори символів, які випадково можуть зійтися з форматом телефонного. Тоді варто використовувати обмеження або шаблони при пошуку телефонних номерів, що може натрапити на попередньо описану проблему з різними форматами номерів;

- якщо текст неочищений від зайвих пробілів або пунктуації, можуть виникнути проблеми зі зчитуванням формату номеру. Для уникнення цього, варто очищувати рядок перед роботою з ним.

6. Чому регулярні вирази ефективні для текстового аналізу?

Регулярні вирази дуже ефективні для текстового аналізу, оскільки вони дозволяють швидко та точно знаходити, змінювати й аналізувати дані. Вони дозволять створювати гнучкі та швидкі засоби для обробки великих масивів даних, при цьому регулярні виразу можна використовувати для будь-якої мови програмування, що значно підвищує універсальність.

7. Як перевірити коректність знайдених номерів?

Щоб перевірити коректність знайдених номерів, можна перевіряти код країни телефонного номеру, якщо пошук обмежений певною територією. Можливо також перевіряти кількість символів у номері та чи всі вони є числами, якщо відома територіальна складова перевірки, адже не кожна країна має номер довжиною 10 символів і кожен код можу відрізнятися за довжиною. Усе це можна здійснити за допомогою регулярних виразів.

8. Яка різниця між методами re.search() і re.findall()?

Метод re.search() зупиняє перевірку по рядку щойно знайде перший збіг до аргумента (регулярний вираз або рядок). Метод re.findall() робить перевірку за аргументом по всьому рядку та не зупиняється, поки не пройде рядок повністю. Всі знайдені збіги він зберігає в списку.

9. Як групуються символи у регулярних виразах?

У регулярних виразах символи групуються за допомогою дужок () та спеціальних конструкцій. Наприклад, регулярний вираз r'(\d{2})-(\d{2})-(\d{4})' розбиває дату на три групи: день, місяць і рік. Якщо для виводу на консоль використати ще й функцію .groups(), воно відобразиться як окремо записані в список числа. Таким чином, за допомогою регулярних виразів групи також можна проіменувати r'(?P<day>\d{2})-(?P<month>\d{2})-(?P<year>\d{4})'. Групувати також можна за допомогою знаку «|», яке означає «АБО».

10. Чому важливо правильно обирати регулярні вирази для аналізу тексту?

Правильний вибір регулярного виразу при аналізі тексту критично важливий, оскільки він впливає на точність, швидкість і надійність обробки даних. Необхідно бути уважним, аби не використати занадто узагальнений вираз, який би призвів до знаходження великої кількості неприйнятних до мотивів збігів, що збільшить час виконання програми та призведе до необхідності додаткової роботи над отриманими даними.