Опис даних, які отримуватиме програма

Як було зазначено у попередньому завданні, дані, які надходять з youtube-transcripts-api, є у форматі json. Ключем у цьому файлі є id відео у youtube, а значенням є список словників, ключами яких є «text», «start», «duration». Тобто субтитри надані не окремим значенням, а розділені залежно від проміжку відео, до якого вони належать. Наприклад, {«iCvmsMzlF7o»: [{«text": "Тож, я почну ось з чого:", "start": 15.26, "duration": 2.0}, {"text": "декілька років тому мені подзвонила організатор заходу,", "start": 17.26, "duration": 2.0}]}. Варто зазначити, що дані про час тут надані у секундах та мілісекундах.

Оскільки мені не важлива інформація про час, до якого стосуються ці субтитри, а потрібен лише текст, то отриманий json-файл зведеться до такого, у якому ключ таки залишиться id відео, а значенням будуть зібрані значення з кожного ключа «text» з отриманого файлу. Наприклад, [{"640BQNxB5mc": "Тож, я почну ось з чого: декілька років тому мені подзвонила організатор заходу,}]

Проте постає питання, звідки дістати ці id відео? Звичайно, я могла б самостійно відбирати TED відео та зберігати їхні id у csv-файлі, але таких відео надзвичайно багато. На щастя, я змогла знайти уже готовий csv-файл на сайті kaggle за посиланням: https://www.kaggle.com/goweiting/ted-talks-transcript#ted_metadata_youtube.csv. Окрім id, тут також зазначений рейтинг відео, кількість «лайків», кількість переглядів та ін. Цю додаткову інформацію я зможу використати як доповнення до знайдених відео для представлення користувачу на кінцевій html сторінці. Для роботи з цим файлом я використовую бібліотеку pandas, щоб прочитати файл та вийняти id з останнього стовпця. Якщо в майбутньому мені потрібно буде отримати значення інших стовпців, я буду користуватися тією ж бібліотекою.

Тобто принцип роботи з даними такий:

1. Опрацювати дані з файлу csv та вибрати id;
2. Використати ці id для того, щоб створити датасет з субтитрами за допомогою youtube-transcripts-api;
3. Виділити потрібну інформацію з отриманого json-файлу та підготувати дані для подальшого дослідження;

Опис можливостей модулів, пакунків модулів, бібліотек, які будуть використовуватися для роботи з даними

Як уже зазначалося, для роботи з даними я буду використовувати такі бібліотеки :

- pandas
- json

З цими бібліотеками мені доводилося працювати й раніше для виконання практичних завдань.

Бібліотека json (https://docs.python.org/3.5/library/json.html#module-json) містить функції та запису та читання файлу, а також є декілька функцій для кодування та декодування файлів. Для реалізації свого проекту я виокристовую функцію dump, яка дозволяє мені записати отримані дані з API у файл. Проте у цій бібліотеці є ще функція dumps, яка не зберігає дані у файл, проте дозволяє виводити їх у терміналі. Оскільки я планую розробити функцію, яка дозволятиме користувачу завантажити субтитри українською, то мені краще використовувати функцію dump.

У бібліотеці pandas втілено багато потрібних мені функцій для роботи з даними. Наразі цю бібліотеку я застосовую для виділення з csv-файлу id відео, яке використовується для завантаження субтитрів за допомогою API. Для цього після використання функції читання файлу *read_csv()*, я створюю дата-фрейм за допомогою функції *DataFrame()*, з якого за ключем «*vidID_youtube*», витягую id відео, до яких вони існують.

За допомогою функції *get()* у мене є можливість виділити кілька потрібних мені стовпців, тому вона мені знадобиться для того, щоб отримати рейтинг відео, кількість переглядів та опис до конкретного відео.

Визначення функціональних та нефункціональних вимог до програми, яка буде розроблятися.

Функціональні вимоги:

1. Високий пріоритет:

- Система дозволяє користувачу знайти перелік відео (не більше 5), увівши ключові слова;
- Користувач отримує короткий опис кожного знайденого відео;
- Користувач отримує кількість переглядів знайдених відео;
- Користувач отримує посилання на сторінку youtube.com;
- Користувач отримує посилання на сторінку ted.com;
- Користувач отримає кількість ted конференцій, субтитри яких написав такий самий перекладач
- Користувач отримає рейтинг поточного відео;
- Користувач не обмежується лише одним пошуком;

2. Середній пріоритет

- Система дозволить користувачу переглянути відео із субтитрами, не переходячи за посиланням;
- Користувач отримає список усіх наявних посилань з українськими субтитрами;
- Користувач отримує список усіх наявних посилань з українськими субтитрами, посортовані за:
    - Рейтингом
    - Кількістю переглядів
    - Тривалістю відео
    - В алфавітному порядку

3. Низький пріоритет:

- Користувач зможе завантажити відео;
- Користувач має змогу завантажити файл з українськими субтитрами;

- Якщо користувач переглядатиме відео локально, то також отримає перелік посилань на конференції, у яких такий самий перекладач;

Нефункціональні вимоги:

- Пошук та виведення потрібних відео не повинні займати більше, ніж 7 секунд;
- Для того, щоб система працювала цілодобово, у будь-яких браузерах та платформах, вона буде розміщена на pythonanywhere.com;
- Розробка буде відбуватися на мові Python;
- Система не зберігатиме ввід користувача;
- Користувач не має змоги вносити зміни;
- Виведення списку наявних посилань з українськими субтитрами не має тривати довше, ніж 1 секунда;
- Вимога до якості втілення пошуку;