Тестове завдання (Python internship)

Ідея

Припустимо, що в є текст, який ви хотіли би зробити більш унікальним. Наприклад, у вас є опис історичного району міста Барселона. Уявіть, що ви хотіли би розмістити його в своєму онлайн-путівнику. Щоб він виводився в топі пошуку, він не має бути занадто схожий на інші подібні тексти, і для цього потрібен мікросервіс, що вміє перефразовувати текст (а саме його синтаксичні дерева) без зміни його сенсу.

Що таке синтаксичне дерево?

Детальніше прочитати тут, але якщо коротко, то це представлення синтаксичної структури тексту у вигляді дерева, де кожний вузол має 2 атрибути:

- Label - синтаксичний тип вузла, що
  - для одного слова є частиною мови. Наприклад, NN - noun (club), DT - determiner (a/an/the/...) тощо.
  - для декількох слів є видом словосполучення. Наприклад, NP - noun phrase, VP - verb phrase тощо.
- Value - значення вузла, що є або списком піддерев, або словом.

Технічне завдання

Створити API на Python, де реалізований єдиний ендпоінт, який приймає на вхід синтаксичне дерево англійського тексту і повертає його перефразовані версії:

- path: `/paraphrase`
- HTTP method: **GET**
- query parameters:
  - **tree**: *str (required)* – синтаксичне дерево у вигляді строки (див. приклад нижче)
  - **limit**: *int (optional, default: 20)* - максимальна кількість перефразованих текстів, що треба повернути response: список перефразованих дерев в форматі JSON

Перефразування має бути зроблене наступним чином:

1. Знайти в тексті всі `NP` (noun phrase) - іменникові словосполучення, що складаються з кількох `NP`, розділених тегами `,` (комою) або `CC` (зв'язковим зворотом, напр. "and").
2. Згенерувати варіанти перестановок місцями цих дочірніх `NP` один з одним.

Для виконання задачі можна використовувати будь-які бібліотеки для роботи з такими деревами – наприклад, nltk.tree. Для створення API теж можна обрати довільний фреймворк (Flask, Django, FastAPI etc).

Вихідний код проекту необхідно розмістити на GitHub, а посилання на нього переслати рекрутеру. Бажано описати в README, як запустити проект для перевірки.

Завдання розраховане на 3-4 години роботи. При рев'ю будуть враховуватись читабельність коду та наскільки легко його можна було б розширювати (наприклад, якщо в майбутньому треба було б додати нові способи перефразування).

Приклад

Для прикладу візьмемо наступний текст з Google Maps: *The charming Gothic Quarter, or Barri Gòtic, has narrow medieval streets filled with trendy bars, clubs and Catalan restaurants*. Його синтаксичне дерево виглядає наступним чином:

```
(S
  (NP
    (NP (DT The) (JJ charming) (NNP Gothic) (NNP Quarter))
    (, ,)
    (CC or)
    (NP (NNP Barri) (NNP Gòtic)))
  (, ,)
  (VP
    (VBZ has)
    (NP
      (NP (JJ narrow) (JJ medieval) (NNS streets))
      (VP
        (VBN filled)
        (PP
          (IN with)
          (NP
            (NP (JJ trendy) (NNS bars))
            (, ,)
            (NP (NNS clubs))
            (CC and)
            (NP (JJ Catalan) (NNS restaurants)))))))))
```

Приклад URL запиту (скопіювати і вставити в браузер, замінивши `<port>` на потрібний):
`localhost:<port>/paraphrase?tree=(S (NP (NP (DT The) (JJ charming) (NNP Gothic) (NNP Quarter)) (, ,) (CC or) (NP (NNP Barri) (NNP Gòtic))) (, ,) (VP (VBZ has) (NP (NP (JJ narrow) (JJ medieval) (NNS streets)) (VP (VBN filled) (PP (IN with) (NP (NP (JJ trendy) (NNS bars)) (, ,) (NP (NNS clubs)) (CC and) (NP (JJ Catalan) (NNS restaurants))))))))`

В файлі `expected-result-example.json` є приклад очікуваного результата виклику ендпоінту (порядок в списку може відрізнятися; в серіалізованих деревах можуть бути переноси та відступи – головне, щоб його потім можна було десеріалізувати, тобто зберіглася загальна структура дужок). Бажаємо успіхів!