BIGDATA TEAM

#W1L1PA: Inverted Index, Starter

Ungraded programming assignment

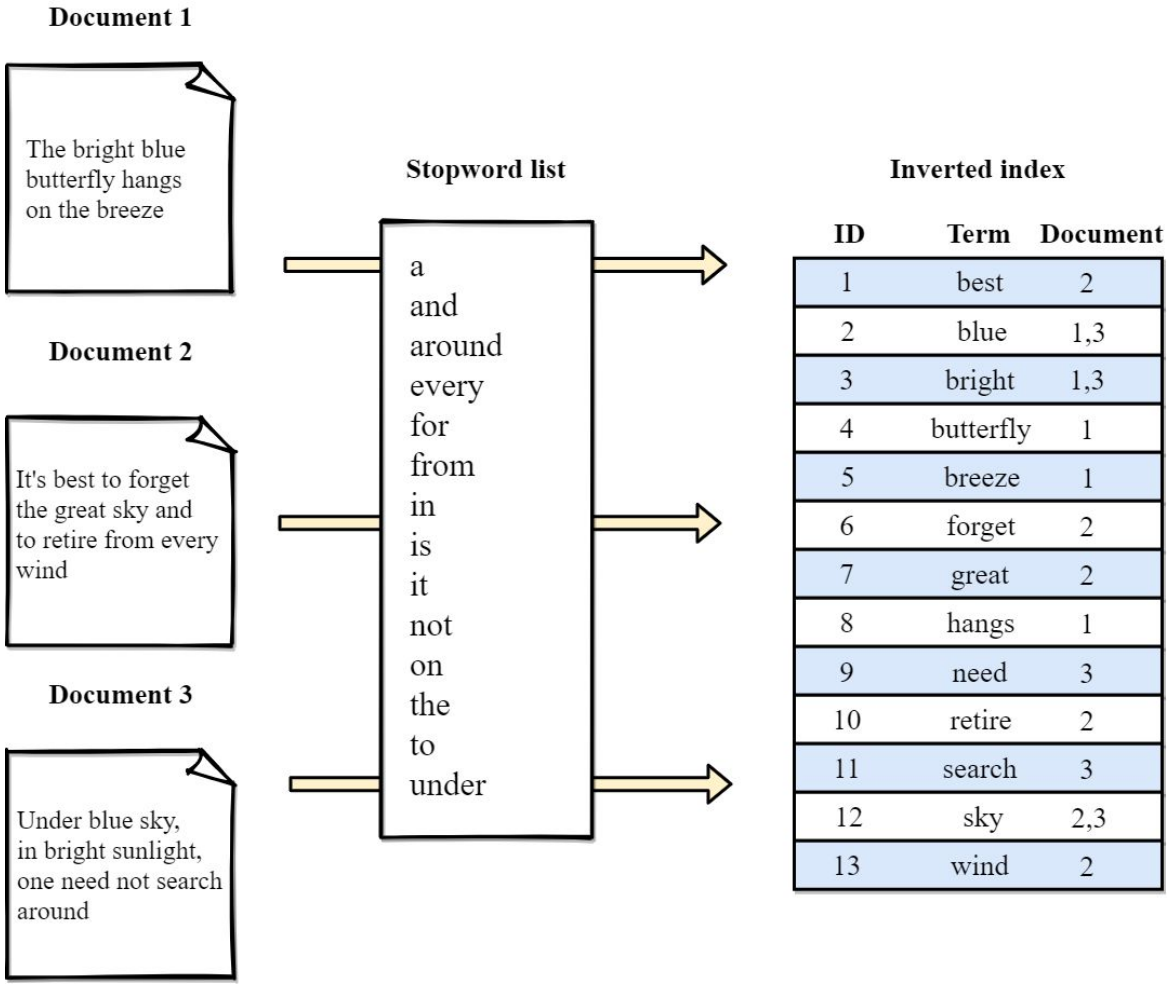
1. Описание задания

В этом задании вам нужно написать библиотеку по построению инвертированного индекса. Цель задания - приучить себя сначала писать тесты, а уже потом реализацию (TDD).

В следующих видео курса будет представлен разбор тестов с примерами. Поэтому, **для лучшего усвоения материала рекомендуется сначала попробовать выполнить это задание самостоятельно**. Очень вероятно, что в период написания тестов у вас появятся вопросы типа “a как лучше сделать x?”. Это нормально, не беспокойтесь, пишите как получается, а лучшие практики разберем чуточку позже.

2. Инвертированный индекс (Inverted Index)

Инвертированный индекс представляет собой словарь, где ключами являются слова (термы), а значениями - списки идентификаторов документов, в которых указанный терм встречается (см. Рис. 1).



(Рис. 1) Инвертированный индекс

Такая структура позволяет поисковым системам найти страницы в интернете, которые могут быть релевантны пользовательскому запросу. Вам будет предоставлен датасет из документов и по этому датасету нужно построить инвертированный индекс. Библиотека должна предоставить возможность:

1. Считать датасет в оперативную память;
2. Разбить каждый документ на термы (слова);
3. ~~Удалить все стоп-слова;~~[1]
4. Построить инвертированный индекс;
5. Сохранить инвертированный индекс на диск;
6. Загрузить инвертированный индекс с диска;
7. Найти документы, соответствующие заданному поисковому запросу (если в запросе указаны слова “Python” и “code”, то нужно вывести только те документы, которые содержат **оба** этих слова).

На Рис. 1 указаны стоп-слова, это слова, которые встречаются практически в каждом документе. В связи с этим, хранение списка документов, в которых встречается это стоп-слово не только практически бессмысленно, но еще и болезненно, поскольку хранение этой информации занимает много места в оперативной памяти (или на жестком диске). Бывают и исключения, иногда нужно находить документы на запросы, которые состоят полностью из стоп-слов, например “to be or not to be”. Но решение этой проблемы выходит за рамки нашего курса. Короткий ответ - можно хранить вместо термов биграммы из стоп-слов (например “to be”, “or not” и т.п.) и искать по ним. Длинный ответ - почитайте релевантную литературу по Information Retrieval (информационному поиску).

Любимая книга автора курса на эту тему:

- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
- https://nlp.stanford.edu/IR-book/ (можно скачать бесплатно)

3. Описание данных

3.1 Дамп Википедии

- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:

[1] В этом задании стоп-слова пока не учитываем, считаем их обычными словами (термами).

1. `INT` - id статьи,
2. `STRING` - текст статьи,

Пример:

```
12    Anarchism         Anarchism is often defined as a political
philosophy which holds the state to be undesirable, unnecessary, or
harmful.
```

3.2 Стоп-слова

- Формат: одно стоп-слово на строчку

Пример:

```
...
wherein
whereupon
wherever
...
```

4. Задания

1. Настройте окружение для разработки:
   - https://github.com/big-data-team/python-course
2. Скачайте необходимые датасеты для выполнения задания:
   - https://github.com/big-data-team/python-course#study-datasets
3. Возьмите за основу Starter реализации библиотеки:
   - https://github.com/big-data-team/python-course/blob/master/inverted_index_starter.py
4. Добавьте пустой test_inverted_index.py и вперед

Вам нужно реализовать объявленные функции в Starter'е. Поскольку инвертированный индекс по факту представляет собой словарь, то его хранение на жестком диске рекомендуется выполнить с помощью стандартной библиотеки json (см. json.load(s) и json.dump(s)). Добейтесь "высокого"[2] уровня покрытия вашего кода тестами (см. pytest --cov и --cov-branch).

5. Изучите PEP-8 и исправьте ваш код (см. также pylint).

[2] Что значит "высокого" - на самом деле предмет для дискуссии в будущих видео. Попробуйте, для начала, определить его для себя самостоятельно.