

#W2PA: Inverted Index CLI

1. Описание задания	2
2. Инвертированный индекс (Inverted Index)	2
3. Описание данных	3
4. Задания	4

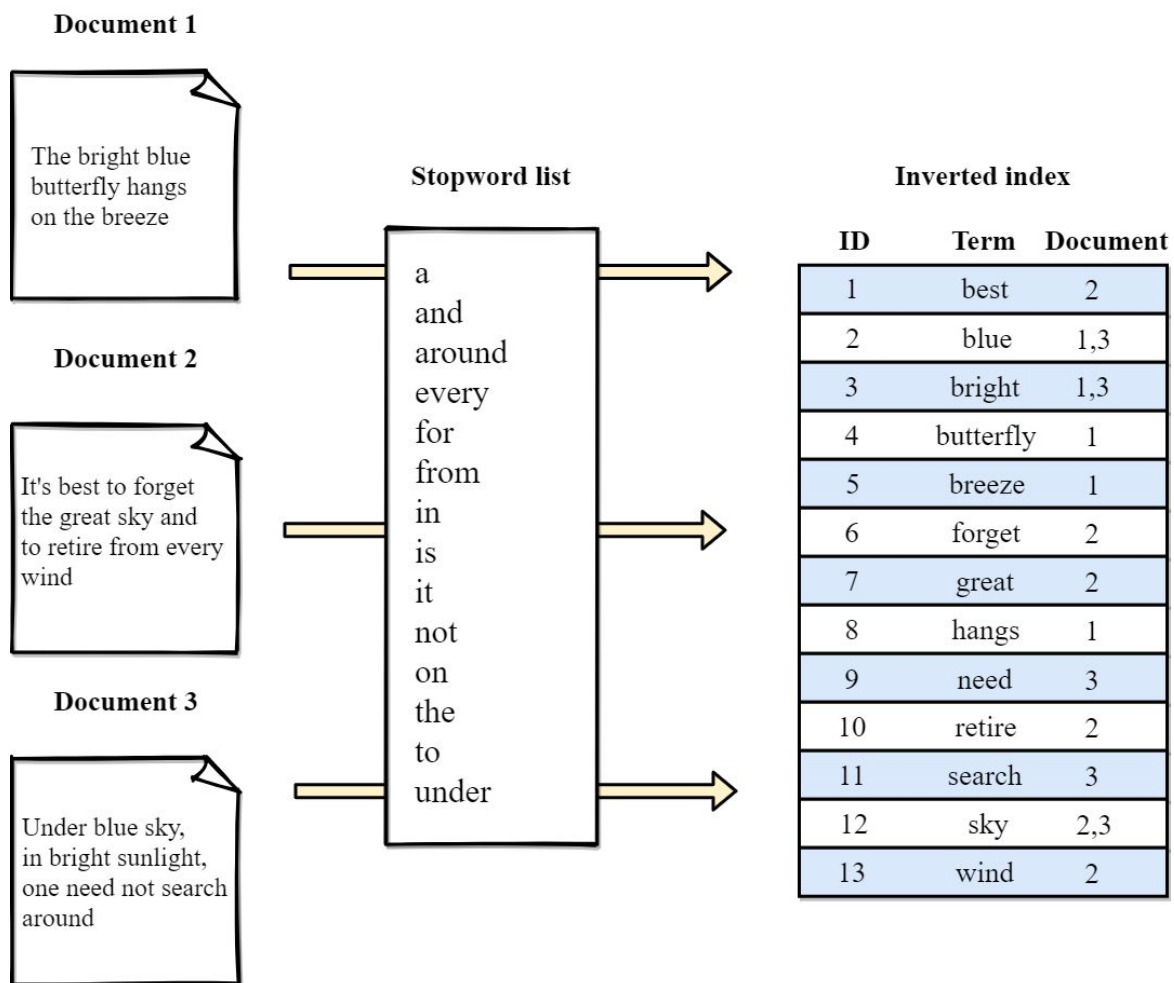
1. Описание задания

В этом задании вам нужно написать консольный интерфейс к библиотеке по работе с инвертированным индексом. Цель задания - завести привычки:

1. Аннотировать код, писать документацию и выбирать лаконичные имена методов и функций (naming);
2. Читать официальную документацию с целью поиска релевантной функциональности.

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. Консольное приложение должно предоставлять возможность:

1. Построить инвертированный индекс и сохранить его на диске используя различные стратегии (см. [argparse:add_argument:choices](#) со значением по умолчанию) - `"inverted_index.py build ..."`;
2. Удалить стоп-слова при построении индекса - `"inverted_index.py build --stop-words <path> ..."`;
3. Найти документы, соответствующие поисковым запросам. Если в запросе указаны слова "Python" и "code", то нужно вывести только те документы, которые содержат **оба** этих слова. (см. `action='append'`) - `"inverted_index.py query --json-index <path> --query <word> [<word> ...] --query <word> [<word> ...] ..."`

3. Описание данных

3.1 Дамп Википедии

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

Prerequisites:

1. Настройка окружения: <https://github.com/big-data-team/python-course>
2. Датасеты: <https://github.com/big-data-team/python-course#study-datasets>

Возьмите за основу решение задания #W1L2PA: Inverted Index Library

С учетом примеров и лайфхаков, показанных на этой неделе:

1. Обновите код¹, чтобы он было более лаконичный (см. collections) и не содержал потенциальных ошибок (наследование, значения по умолчанию для сложных объектов и т.п.);
2. Изучите [PEP-257](#) (в отличие от PEP-8 - очень короткий) и дополните вашу библиотеку документацией. Запустите pylint **без** указания флагов “-d invalid-name,missing-docstring”, убедитесь, что ошибок и предупреждений нет;
3. Изучите документацию по [argparse](#), чтобы выполнить задания 1, 2 и 3 из раздела “2. Инвертированный индекс (Inverted Index)”;
4. Поскольку это задание для самостоятельной работы, то поделитесь в канале группы информацией и фидбеком по факту выполнения задания: #W2PA #inverted_index #cli #complete. Не забываем отслеживать уровень покрытия тестами.

¹ В какой шляпе делаем рефакторинг? Красной или зеленой?