**Московский авиационный институт  
(национальный исследовательский университет)**

Факультет прикладной математики и физики  
Кафедра вычислительной математики и программирования

**Лабораторная работа №8  
по курсу «Информационный поиск»**

Ранжирование TF-IDF

|  |  |  |
| --- | --- | --- |
|  | **Студент:** | Рожлейс И.А. |
| **Группа:** | М8О-108М-18 |
| **Преподаватель:** | Калинин А.Л. |
| **Дата:** | 09.06.2019 |

Москва, 2019

**Цель работы**

Необходимо сделать ранжированный поиск на основании схемы ранжирования TF-IDF. Теперь, если запрос содержит в себе только термины через пробелы, то его надо трактовать как нечёткий запрос, т.е. допускать неполное соответствие документа терминам запроса и т.п. Примеры запросов:

• [ роза цветок ]

• [ московский авиационный институт ]

Если запрос содержит в себе операторы булева поиска, то запрос надо трактовать как булев, т.е. соответствие должно быть строгим, но порядок выдачи должен быть определён ранжированием TF-IDF. Например:

• [ роза && цветок ]

• [ московский && авиационный && институт ]

В отчёте нужно привести несколько примеров выполнения запросов, как удачных, так и не удачных.

**Ход работы**

Для выполнения этой работы потребовалась модернизация лабораторной работы №3 «Булев индекс». Теперь помимо прямого и координатного индексов программа собирает статистику корпуса документов и сохраняет её в файл. Далее этот файл используется для ранжирования по формуле TF-IDF.

Статистический файл, как и индексные файлы поделён на три секции: заголовок файла, словарь и секция данных. Структурная схема файла представлена на **табл. 1**. Заголовок файла имеет такие поля данных как количество уникальных токенов (порядка 1,5 млн.) и количество документов в корпусе (порядка 65 тыс.) Далее следует словарь файла.

Словарь состоит из записей. Одна запись описывает один токен. В записи указан идентификатор токена, количество документов корпуса в которых встречается токен и смещение в байтах относительно начала файла в секцию данных где расположен список документов, содержащих этот токен. Идентификаторы токенов представляют собой контрольную сумму байт UTF-8 строки токена по алгоритму CRC-32.

Секция данных представляет собой список документов и информацию, связывающую токен с документом: количество вхождений токена в документ и количество всего токенов в документе. Информация с секции данных избыточна. Так, например, количество всего токенов в документе не зависит от количества вхождений конкретного токена, тем не менее эта информация дублируется для каждого документа. Файл возможно хранить более компактно, однако в таком случае возрастает сложность работы с файлом.



**Табл. 1.** Структурная схема статистического файла.

Вес документа высчитывается как сумма меры TF-IDF токенов входящих в состав документа, которая представляет собой произведение множителей TF и IDF:

Мера TF (term frequency) определяет частоту вхождения токена в документ и представляет собой отношение количества вхождений токена к общему количеству токенов в документе:

где n количество вхождений токена, а N количество всего токенов в документе. Чем чаще употребляется искомый токен в документе, тем больший вес имеет этот документ.

Мера IDF (inverse document frequency) в свою очередь определяет частоту использования токена среди документов всего корпуса и по своей сути перераспределяет значимость документов в сторону редкоупотребляемых токенов. Так чем в меньшем количестве документов упоминается искомый токен, тем больший вес имеют эти документы:

где N общее количество документов в корпусе, а d количество документов в состав которых входит токен.

Ранжирование списка документов происходит в функции range посредством сортировки документов по их весу.



Сначала функция формирует вектор пар *<документ, токен>* для которых считывает информацию из файла статистики. Для каждой пары высчитывается мера TF-IDF, суммируется и присваивается нужному документу. Далее список пар *<документ, вес>* сортируется функцией стандартной библиотеки std::sort по ключу *«вес»*. Способ сортировки описан предикатом pred который передаётся третьим аргументом.

**Заключение**

Был рассмотрен способ ранжирования документов по формуле TF-IDF. Был разработан формат файла для хранения статистической информации корпуса документов. Размер статистического файла равен 189 Мб.

Было оценено качество поисковой выдачи по P-показателю (precision) который составил 0.4, 0.5, 0.5 на уровнях @1, @3 и @5. Аналогичная характеристика была вычислена в лабораторной работе №2 «Оценка качества поиска» для поисковика Википедии, что даёт возможность сравнить собственное качество поиска с поиском Википедии метрика которого на тех же уровнях оказалась равной 0.11, 0.23 и 0.36.

|  |  |  |
| --- | --- | --- |
|  | **Запрос** | **P** |
| **1** | «фотосинтезирующие бактерии» | 0 0 1 0 1 |
| **2** | «хладнокровные рептилии» | 1 1 1 1 1 |
| **3** | «палочка Коха» | 0 1 0 0 0 |
| **4** | «аэробные бактерии» | 0 1 0 1 1 |
| **5** | «Дарвин» | 1 1 1 0 1 |
| **6** | «дезоксирибонуклеиновая кислота» | 1 0 0 0 0 |
| **7** | «бубонная чума» | 0 0 1 0 0 |
| **8** | «бывает ли так что рыбы летают» | 0 0 0 0 0 |
| **9** | «фотосинтез» | 1 0 1 1 1 |
| **10** | «рецессивный признак» | 0 1 1 1 0 |

Из таблицы запросов видно, что односложные запросы выполняются очень хорошо. Двух и более сложные запросы отрабатываются сравнительно неплохо, однако в начало списка попадают документы имеющие мало общего с изначальным запросом. Запросы по типу вопросов обрабатываются очень плохо, так как обработка такого рода запросов требует наличия специальных алгоритмов.

**Источники**

[1] kshiian. Алгоритм BM25 [Электронный ресурс] // URL: <https://habr.com/ru/post/162937/>