Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №6 по курсу «Информационный поиск»

Студент: В. М. Филиппов
Преподаватель: А. А. Кухтичев
Группа: М8О-410Б

Дата:
Оценка:
Подпись:

Москва, 2025

Лабораторная работа №5 «TF-IDF»

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

1 Описание

В рамках этой лабораторной работы необходимо доработать поисковый движок, так чтобы он мог ранжировать результаты поиска по метрике TF-IDF

1 TF-IDF

TF-IDF — это статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции или корпуса.

В отличие от булева поиска, который просто говорит «да» или «нет», TF-IDF позволяет ранжировать документы, выделяя те, где искомое слово является наиболее значимым.

Метрика состоит из двух множителей:

- TF (Term Frequency) — частота слова, которая определяет, насколько часто слово встречается в конкретном документе.

$$TF(t,d) = \frac{n_{td}}{\sum_k n_{kd}} \tag{1}$$

  Или можно вот так. Это необходимо, чтобы длинные документы, в которых много раз может встречаться одно и тоже слово, не портили выдачу.

$$wf_{td} = \begin{cases} 1 + \log(tf_{td}), & \text{если } tf_{td} > 0 \\ 0, & \text{в противном случае} \end{cases} \tag{2}$$

- IDF (Inverse Document Frequency) — Обратная частота документа, которая снижает вес слов, которые встречаются слишком часто во всех документах (например, «и», «в», «что», «который»).

$$IDF(t,D) = \log\left(\frac{N}{|\{d \in D : t \in d\}|}\right) \tag{3}$$

Итоговая формула TF-IDF.

$$\text{TF-IDF}(t,d,D) = TF(t,d) \cdot IDF(t,D) \tag{4}$$

2 Исходный код

Для реализации TF-IDF метрики применяется следующее: для каждого токена в документе при добавлении его в индекс мы считаем сколько раз он в нём встретился. Это будет метрика TF. При дампинге на диск в постинг списке теперь будет храниться пара (doc_id, tf).

При поиске алгоритм такой же (получаем токены из запроса, формируем ОПН, получаем документы, которые удовлетворяют нашему запросу) до момента возврата результата поиска. В методе processResult вызывается метод rankResults, внутри которого происходит следующее:

1. В цикле для каждой термы запроса мы считаем IDF

$$IDF(t) = \ln\left(\frac{N}{1 + df_t}\right) \tag{5}$$

   IDF здесь я высчитывал таким образом, чтобы в случае, если токен встречается в каждом документе, это не убило метрику (IDF будет 0 в таком случае).

2. Для каждого документа в постинг списке данного токена проверяем, находится ли этот документ в том, что выдала булева часть алгоритм, если да, то добавляем этому документу score равный $tf * idf$

После того, как каждому документу в выдаче назначен рейтинг, результаты запроса сортируются согласно этому рейтингу.

1 Примеры запросов

При запросе manchester | united поисковик выдал в топе титульную статью с Википедии. Я очень сильно удивился, когда увидел, что в топе выдачи статья на футболиста Джона Айткена, но перейдя по ссылке я понял, что это редирект на статью со списком игроков МЮ за всю историю. 3 ссылка это опять же редирект на статью про Уэйна Рунни, легенду манчестерского футбола.

```
Enter query: manchester | united
Tokens searching: [manchest, unit, ]
Top 10 results
[1] https://en.wikipedia.org/wiki/John_Aitken_(footballer,_born_1870) TF-IDF: 30.3702
[2] https://en.wikipedia.org/wiki/Manchester_United_F.C. TF-IDF: 29.7001
[3] https://en.wikipedia.org/wiki/Wazza_(footballer) TF-IDF: 28.2861
[4] https://en.wikipedia.org/wiki/Timeline_of_English_football TF-IDF: 27.2021
[5] https://en.wikipedia.org/wiki/Ryan_Giggs TF-IDF: 26.5246
[6] https://en.wikipedia.org/wiki/Old_Trafford TF-IDF: 25.9452
[7] https://en.wikipedia.org/wiki/2023%E2%80%9324_in_English_football TF-IDF: 25.3764
```

```
[8] https://en.wikipedia.org/wiki/Football_records_and_statistics_in_England TF-IDF:
    24.9874
[9] https://en.wikipedia.org/wiki/2018%E2%80%9319_in_English_football TF-IDF: 24.9476
[10] https://en.wikipedia.org/wiki/Mason_Greenwood TF-IDF: 24.6653
Query time: 0.0286911 sec
Number of results: 99459 items
```

В этом запросе я попытался найти что-то про неолимпийские медали за теннис или баскетбол. В итоге я получил статьи со всякими разными наградами, в числе которых есть и награды за теннис, и за баскетбол.

```
Enter query: !olympic & medals & (tennis | basketball)
Tokens searching: [olymp, med, tenni, basketbal, ]
Top 10 results
[1] https://en.wikipedia.org/wiki/List_of_awards_named_after_people TF-IDF: 38.679
[2] https://en.wikipedia.org/wiki/Southeastern_Conference_Athlete_of_the_Year TF-IDF:
    32.5334
[3] https://en.wikipedia.org/wiki/Southeastern_Conference TF-IDF: 32.5334
[4] https://en.wikipedia.org/wiki/List_of_multiple_SEA_Games_medalists TF-IDF: 30.988
[5] https://en.wikipedia.org/wiki/Sporting_goods TF-IDF: 29.3921
[6] https://en.wikipedia.org/wiki/Sports_equipment TF-IDF: 29.3921
[7] https://en.wikipedia.org/wiki/Althea_Gibson TF-IDF: 28.7204
[8] https://en.wikipedia.org/wiki/John_Lucas_II TF-IDF: 27.788
[9] https://en.wikipedia.org/wiki/West_Virginia_Sports_Hall_of_Fame TF-IDF: 27.7662
[10] https://theguardian.com/sport/live/2024/sep/07/paris-2024-paralympics-day-10-
    cycling-canoeing-tennis-athletics-and-more-live TF-IDF: 27.0776
Query time: 0.0167105 sec
Number of results: 1489 items
```

Здесь я попытался сделать так, чтобы мне рассказали именно про спортсменов, убрав из выдачи футбол, чтобы не найти списки общих наград. У меня получилось, почти все статьи рассказывают нам о спортсменах.

```
Enter query: !olympic & medals & (tennis | basketball) & !football
Tokens searching: [olymp, med, tenni, basketbal, footbal, ]
Top 10 results
[1] https://en.wikipedia.org/wiki/List_of_multiple_SEA_Games_medalists TF-IDF: 30.988
[2] https://en.wikipedia.org/wiki/Althea_Gibson TF-IDF: 28.7204
[3] https://en.wikipedia.org/wiki/John_Lucas_II TF-IDF: 27.788
[4] https://en.wikipedia.org/wiki/E._Lilyan_Spencer TF-IDF: 27.0169
[5] https://en.wikipedia.org/wiki/Wii_Sports TF-IDF: 26.8262
[6] https://en.wikipedia.org/wiki/List_of_Canada_Games TF-IDF: 25.3585
[7] https://en.wikipedia.org/wiki/FIBA_U16_EuroBasket TF-IDF: 25.2295
[8] https://en.wikipedia.org/wiki/Tennis_elbow TF-IDF: 24.8621
[9] https://en.wikipedia.org/wiki/Doping_in_tennis TF-IDF: 24.5384
[10] https://en.wikipedia.org/wiki/4-point_player TF-IDF: 22.8808
Query time: 0.00937442 sec
Number of results: 872 items
```

Поисковик почти не выдаёт статьи с theguardian. Например по запросу "manchester

united win arsenal"ссылка на новостной портал встречается лишь на 36 месте. Во-первых, вероятно это связано с непропорциальностью корпуса документов: статей Википедии гораздо больше, чем TheGuardian. Во-вторых, статьи на Википедии длиннее и из-за этого портят метрику TF, даже несмотря на то, что я попытался это скомпенсировать логарифмом.

Время поиска слегка увеличилось из-за того, что приходится сортировать документы по релевантности.

3 Выводы

При выполнении данной лабораторной работы, я узнал про то, как поисковики могут ранжировать результаты. Также я узнал про плюсы и минусы конкретной метрики TF-IDF. Мною был реализован собственный поисковик, использующий инвертированный индекс. В ходе разработки я столкнулся с особенностями подсчета TF и пришел к выводу о необходимости выбора между "сырым"подсчетом вхождений и их логарифмированием для сглаживания весов длинных документов.

Список литературы

[1] Маннинг, Рагхаван, Шютце *Введение в информационный поиск* — Издательский дом «Вильямс», 2011. Перевод с английского: доктор физ.-мат. наук Д. А. Клюшина — 528 с. (ISBN 978-5-8459-1623-4 (рус.))

[2] Understanding TF-IDF (Term Frequency-Inverse Document Frequency) [Электронный ресурс] // GeeksForGeeks. — URL: `https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/` (дата обращения: 18.12.2025).