Лабораторная работа №4

Реализация алгоритма индексирования документов и поиска по индексу.

Цель работы: научиться реализовывать на выбранном языке программирования алгоритмы индексирования документов и осуществлять поиск по индексу.

Краткие теоретические сведения

Инвертированный индекс — структура данных, в которой для каждого слова коллекции документов в соответствующем списке перечислены все места в коллекции, в которых оно встретилось. Инвертированный индекс используется для поиска по текстам.

Булев поиск опирается на использование инвертированного индекса ключевых слов, т. е. таблицы, в которой для каждого ключевого слова перечисляются все документы, где оно встречается. Главным достоинством этого алгоритма является возможность связывания слов запроса логическими операциями. К недостаткам этого алгоритма следует отнести невозможность определения релевантности запросу полученной выборки документов и, как следствие, невозможность ее сортировки.

Опишем как решается задача нахождения документов в которых встречаются все слова из поискового запроса. При обработке однословного поискового запроса, ответ уже есть в инвертированном индексе — достаточно взять список соответствующий слову из запроса. При обработке многословного запроса берутся списки, соответствующие каждому из слов запроса и пересекаются.

Пример.

Пусть у нас есть корпус из трех текстов T_0 = "it is what it is", T_1 = "what is it" и T_2 = "it is a banana", тогда инвертированный индекс будет выглядеть следующим образом:

"a": {2}
"banana": {2}
"is": {0, 1, 2}
"it": {0, 1, 2}
"what": {0, 1}

Здесь цифры обозначают номера текстов, в которых встретилось соответствующее слово. Тогда отработка поискового "what is it" запроса даст следующий результат $\{0,1\} \cap \{0,1,2\} \cap \{0,1,2\} = \{0,1\}$.

Таблицы индекса

Для эффективной организации поиска документов необходимо задействовать несколько таблиц в базе данных. В самом простом случае используются следующие три.

Таблица документов Documents. В ней хранится информация обо всех документах, проиндексированных системой, а именно название документа, его авторы, тип файла, путь к файлу/URL и т. д. При этом каждому документу необходимо присвоить уникальный идентификатор Doc_id.

Таблица ключевых слов/словарь Words. Здесь хранятся все ключевые слова системы и соответствующие им номера Word_id.

Инвертированный индекс Inverse, используемый для поиска. В этой таблице хранится идентификатор слова Word_id и соответствующий ему список документов, содержащих это слово.

Эффективная организация словаря

Одна из самых важных и трудных проблем индексации текстов связана с созданием и пополнением словаря ключевых слов. Главная сложность ее заключается в том, что для эффективной работы системы необходимо рассматривать только базовые словоформы ключевых слов..

Еще одна проблема индексирования связана с выявлением и удалением из текста так называемых стоп-слов. Они не несут смысловой нагрузки в текущей предметной области, и для эффективной работы системы их следует удалять при индексировании. Как правило, стоп-словами являются предлоги, союзы, артикли, вводные слова и т. п. Они очень часто встречаются в документах, но малоинформативны. Для их удаления можно либо использовать отдельный словарь стоп-слов, либо считать все слова с высокими частотами встречаемости в базе данных текстов стоп-словами и удалять их при индексировании.

Ранжирование

Обычно в поисковых системах после построения с помощью инвертированного индекса списка документов, содержащих слова из запроса, идет ранжирование документов из списка.

Для каждого запроса необходимо вычислить значение Score документа – показатель релевантности документа запросу, на основании которого и производится ранжирование.

Для расчета Score предлагается использовать аддитивную модель. В качестве слагаемых в данной модели предлагаются следующие: встречаемость слов из запроса в документе (W_{single}), встречаемость пар слов из запроса в документе (W_{pair}) и встречаемость текста запроса целиком (W_{Phrase}). Помимо этого есть два слагаемых, дающих преимущество за наличие всех слов запроса в документе ($W_{AllWords}$).Итоговая формула выглядит следующим образом:

$$Score = W_{single} + W_{pair} + k_1 * W_{AllWords} + k_2 * W_{Phrase}$$

где $k_1 = 1$, $k_2 = 1/350$

Задание к лабораторной работе

Написать программу на выбранном языке программирования, реализующую индексацию документов (не менее 5 документов) описанным выше алгоритмом и осуществить поиск документа, удовлетворяющего заданный запрос. Осуществить ранжирование документов по их релевантности запросу. Программа должна запрашивать имена входных файлов и выводить заголовки и выдержки из найденных по запросу документов.

Контрольные вопросы

1. Что такое индекс?
2. Что такое инвертированный индекс?
3. Что такое прямой индекс?
4. Принципы работы алгоритма инвертированного индекса.
5. Какие форматы файлов потенциально пригодны для индексирования?
6. Что такое таблица индекса?
7. Как определять релевантность документа?
8. Что такое ранжирование документа?
9. На основе каких параметров осуществляется ранжирование документов?
10. Особенности булевого поиска.
11. Особенности вероятностного поиска.