ПРОБЛЕМЫ И ОСОБЕННОСТИ ПОСТРОЕНИЯ ИНВЕРТИРОВАННОГО ИНДЕКСА КОЛЛЕКЦИИ ДОКУМЕНТОВ

Луда С. Э.
Национальный технический университет
«Харьковский политехнический институт»
г. Харьков, ул. Херсонская, д.12, кв.1
+380937276595, e-mail: *simplement.sy@gmail.com*

Информационный поиск может осуществляться по разным алгоритмам, но именно алгоритм инвертированных индексов сегодня используется всеми крупными поисковыми системами в мире. В чем же заключается процесс индексации коллекции документов?

При использовании алгоритма обратных индексов, поисковые системы преобразовывают документы в инвертированные текстовые файлы. Каждый такой файл представляет собой индексную структуру, состоящую из двух частей:

1 – словарь, содержащий термины, дополнительные структуры, обеспечивающие быстрый поиск по термину;

2 – пост-листы. Каждый пост-лист представляет собой массив адресов вхождений слова - идентификатор документа, или идентификатор документа и позиций слова в документе, или дополнительные флаги и форматирования слова и т.п. Каждый список словопозиций упорядочен по идентификаторам документа.

В ходе анализа был разработан прототип системы для индексации коллекции документов на языке C++ в среде «Borland C++ Builder». Он производит индексацию трех текстов: «Document Indexing Tutorial.txt», «Introduction to IR.txt», «IR Wikipedia.txt». Рассмотрим алгоритм построения инвертированного индекса этой коллекции.

На первом этапе программа размечает текст, с помощью функции strtok() разбивая его на лексемы - последовательности символов, объединенные в семантическую единицу для обработки.

На следующем этапе осуществляется нормализация лексем – это процесс приведения лексем к канонической форме, осуществляемый для устранения несущественных различий между последовательностями символов. В результате мы получаем список терминов. Термин – это нормализированная лексема, включенная в словарь системы информационного поиска.

В данной программе все лексемы приведены к нормализованному виду терминов вручную, так как в нашем примере достаточно разделить текст по пробелам и отбросить знаки пунктуации. Однако при обработке большой коллекции документов возникает необходимость решения более сложных задач, таких как установление функций дефиса или апострофа, а также пробела, т.к. некоторые слова, разделенные пробелом, семантически обозначают одну лексему (New York, Los Angeles). Некоторые системы игнорируют т.н. стоп-

слова (stop-words) - распространенные слова, не представляющие ценности для удовлетворения информационных потребностей пользователей, например, a, and, be, for, has, he, is, it, to, и др. Однако при поиске фразы ее смысл может быть утерян, если не индексировать эти слова. Так, например, некоторые фразы целиком состоят из стоп-слов («To be or not to be», «Let it be»).

Часто в поисковых системах игнорируется регистр букв (case-folding). Но это может привести к непреднамеренному расширению запроса, т.к. многие имена собственные отличаются от имен нарицательных лишь прописной буквой. Примером являются General Motors, The Fed(Федеральная резервная система), фамилии Bush, Black.

Следующая проблема состоит в том, что по грамматическим причинам в документе встречаются разные формы одних и тех же слов. Существует два способа решения – это стемминг и лемматизация. *Стемминг* – это процесс, в ходе которого от слов отбрасываются окончания с расчетом на то, что в большинстве случаев это себя оправдает. *Лемматизация* – это точный процесс с использованием лексикона и морфологического анализа слов, в результате которого удаляются только флективные окончания и возвращается основная форма слова, называемая *леммой*. Например, лексема «saw» в ходе стемминга может превратиться в букву «s», а лемматизация вернет либо слово «see», если эта лексема является глаголом, либо «saw», если она – имя существительное. Несмотря на то, что для некоторых процессов лемматизация может оказаться очень полезной, для остальных запросов она существенно снижает производительность. Стемминг же повышает полноту, но снижает точность поиска.

На последнем этапе программа индексирует документы, в которых встречаются термины, создавая инвертированный индекс, состоящий из словаря и словопозиций. В результате мы получаем таблицу, в которой все термины, содержащиеся в коллекции, расположены в алфавитном порядке. Напротив каждого термина указана частота, с которой он встречается в каждом из документов коллекции отдельно и его общая частота во всей коллекции.

Структура обратных индексов подобна глоссарию книги, в котором указано, где найти документ. Сама идея инвертированного индекса практически не имеет конкурентов, поскольку является наиболее эффективной для текстового поиска по произвольному запросу.