У даній статті представлена розроблена авторами система індексування повнотекстових документів, яка вирішує завдання інтелектуального пошуку інформації за ключовими словами з урахуванням морфологічних особливостей російської мови. Система проста у використанні та має дружній інтерфейс користувача. Система може використовуватися в автоматизованих інформаційних бібліотечних системах, системах автоматичного реферування

Ключові слова: повнотекстовий пошук, інтелектуальні системи, індексація, морфологічний аналіз, автоматизація бібліотечної діяльності

В данной статье представлена разработанная авторами система индексирования полнотекстовых документов, решающая задачу интеллектуального поиска информации по ключевым словам с учетом морфологических особенностей русского языка. Система проста в использовании и обладает дружественным пользовательским интерфейсом. Система может использоваться в автоматизированных информационных библиотечных системах, системах автоматического реферирования

Ключевые слова полнотекстовый поиск, интеллектуальные системы, индексация, морфологический анализ, автоматизация библиотечной деятельности

УДК 004.934

ИНДЕКСИРОВАНИЕ ПОЛНОТЕКСТОВЫХ ДОКУМЕНТОВ ДЛЯ ЗАДАЧИ ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА ИНФОРМАЦИИ ПО КЛЮЧЕВЫМ СЛОВАМ

Н. В. Борисова
Ассистент*
E-mail: borisova_nv@mail.ru
З. А. Кочуева
Старший преподаватель*
*Кафедра интеллектуальных компьютерных систем
Национальный технический университет
"Харьковский политехнический институт"
ул. Фрунзе, 21, г. Харьков, Украина, 61002
E-mail: kochueva@kochuev.com

1. Введение

Актуальность полнотекстового поиска сегодня обусловлена необходимостью производить поиск информации в постоянно увеличивающемся объеме электронных документов. Очень важно уметь быстро находить в этом объеме действительно нужную информацию. Актуальными также являются всевозможные средства интеллектуализации поиска, которые разрабатываются, как правило, без учета морфологии естественного языка [1]. При создании эффективных информационно-поисковых систем, использующих в качестве критериев поиска набор ключевых слов, одной из проблем является индексирование текстов. Задача состоит в том, чтобы «правильно» определить этот набор [2].

2. Литературный обзор исследований и постановка проблемы

На сегодняшний день существует довольно много различных вариантов поиска текстов (или их фрагментов) по ключевым словам, каждый из них имеет свои достоинства и недостатки.

Работа поисковых систем часто основана на использовании ключевых слов, что подразумевает возможность выделения из каждого документа некоторого текстового содержания. Существуют форматы документов, извлечь текстовое содержание из которых непросто. Важна информация о названии документа и его авторах, она часто рассматривается отдельно от содержания. Ниже приведен список форматов файлов, потенциально пригодных для индексации, с анализом механизма получения их текстового содержания [3].

Текстовые файлы (txt). Это самый удобный для индексации формат, и извлечь текстовое содержание из таких файлов достаточно просто. Данный формат документа наиболее часто применяется в автоматизированных библиотечных системах. Необходимо отметить, что в файлах текстового формата затруднено определение названия документа и практически невозможно выделение его авторов.

HTML-страницы (htm, html). Наиболее распространенный формат хранения текстовой информации, формат HTML относительно легко подвергается обработке для выделения текста.

Документы Adobe Acrobat (pdf). Данный формат получил в последнее время широкое распространение отчасти благодаря своей межплатформенности. Несмотря на то, что pdf-файлы, как правило, содержат текст, для получения его в явном виде обычно требуются значительные усилия. Для этого существует ряд программных продуктов, реализующих полный

разбор pdf-файла. Возможно также использование средств automation Adobe Acrobat, однако из-за больших затрат на межпроцессные вызовы (COM-сервер Adobe Acrobat реализован в виде локального, т. е. exe-файла), процедура получения текста даже для одного pdf-файла требует значительных затрат времени, и, на взгляд авторов, практически неприемлема.

Файлы PostScript (ps). Файлы этого формата также приобрели большую популярность и используются, в частности, для хранения научных статей, чему способствует возможность легкого преобразования в этот формат dvi-файлов.

Документы MS Word (doc). Фирма Microsoft официально не открыла формат doc и не предоставила удобных средств получения содержимого doc-файлов. Тем не менее, с помощью имеющихся автоматических средств данная задача может быть решена, хотя и со значительным ущербом для надежности и скорости работы приложений.

Файлы RTF (rtf). Этот формат был также разработан Microsoft, его описание можно найти на официальном сайте корпорации.

Файлы мультимедиа (mp3, ogg, avi, mpeg и др.). Они могут быть проиндексированы по названию песни, фильма, альбома и имени исполнителя, если эти данные в них присутствуют.

Исполняемые файлы (exe). Иногда такие файлы можно подвергнуть индексации. Хотя текстовое содержание из exe-файлов получить невозможно, но этот формат позволяет определить название программы и ее авторов из ресурсов.

Для файлов, форматы которых не позволяют проводить их прямую индексацию, возможно составление файлов описания (для них часто используется расширение diz) и при индексации получение информации из них. Такой метод полезен при индексации программных продуктов. Он позволяет хранить для каждого из них, помимо названия и фирмы-производителя, аннотацию, что заметно повышает точность навигации.

Проблема индексирования текстов состоит в том, что от ключевых слов (индексов) требуется соблюдение двух взаимоисключающих принципов [4]:

- ключевые слова должны как можно точнее идентифицировать текст;

- ключевые слова должны как можно более точно отражать содержание (смысл) текста.

Рассмотрим каждый из этих принципов. Известно, что определенные ключевые слова полностью идентифицируют текст в заданном подмножестве текстов. Из этого автоматически вытекает, что выбраны такие ключевые слова, которые не встречаются ни в каком другом тексте, кроме того, который они определяют. Понятно, что такими словами могут быть только специфические термины, фамилии авторов, названия каких-то малоизвестных фирм и т. п. Определив, таким образом, ключевые слова, пользователь информационно-поисковой системы должен обязательно их помнить.

Но, как правило, пользователь не может запомнить какие-то крайне редкие термины и хочет видеть в списке ключевых слов те, которые, по его мнению, отражают смысл текста. Очевидно, что редкие термины далеко не всегда являются центральными в тексте, хотя и полностью его идентифицируют. Отсюда получаем противоречие со вторым принципом.

Рассмотрим теперь другой случай. Пусть ключевые слова полностью отражают смысл текста. Но тогда вероятность получения только какого-то одного требуемого текста сильно снижается, поскольку текстов, сходных по смыслу в заданном подмножестве текстов, может быть несколько. Противоречие с первым принципом. Кроме того, остается неясным, как отобрать те ключевые слова, которые полностью отражают смысл текста.

В общем случае эта проблема однозначно не разрешима, хотя и существуют достаточно эффективные системы поиска (например, поисковые системы в Internet) [5]. Однако автоматическое индексирование и поиск ключевых слов в полнотекстовых документах необходимо проводить не только в Internet, но и в современных библиотеках, которые нарастающими темпами накапливают неструктурированные текстовые ресурсы. Причем объем накопленной текстовой информации может быть таким затруднительным, что задача подготовки их полного библиографического описания становится крайне затруднительной. Очевидна необходимость применения специальных решений, которые позволят работнику библиотеки автоматизировать процесс обработки полнотекстовых документов [6].

Распространено мнение, что шаблон "*", означающий любой набор символов, достаточен для поиска в русских текстах. То есть все проблемы, связанные с особенностями морфологии, решаются путем обеспечения развитого языка запросов [7].

Допустим, пользователь пытается найти «ветер в поле». Чтобы найти эту информацию с использованием шаблонов, вероятно, надо ввести слова "ветер" и "поле". Однако, если в тексте были словоформы "ветра" или "полях", остается вариант шаблона – "вет*" и "пол*", что приведет к включению в результаты поиска материалов о «польской ветчине» и «политике вето». Отсутствие морфологии сильно, а иногда катастрофически влияет на чувствительность и избирательность поиска. Запрос "ветер И поле" уменьшает чувствительность, а "вет* И пол*" – избирательность.

Случаи, когда шаблон не позволяет достичь приемлемых результатов:

– слова, у которых в разных формах меняется основа (супплетивные формы): *класть – положить, идти – шел, плохо – хуже, я – меня, человек – люди, ребенок – дети*;

– слова с большим количеством словоформ. Привести список всех словоформ русского глагола (с причастными и деепричастными формами - до 250 различных форм) человеку, не имеющему лингвистического образования, очень трудно. Понять, все ли словоформы из этого списка "накрываются" шаблоном "*" – ещё труднее;

– слова с беглыми гласными и чередованиями. В русском языке, примерно четверть слов имеет чередования, которые не позволяют найти слово по шаблонам: (*искать – ищу, окно – окон, расти – рос*: запрос и* или ок* или р* даст много информации, не относящейся к предмету поиска);

– короткие (три-четыре буквы) слова: *дом, хор*, и т.п. Во всех языках, в том числе и в русском, имеется

общая закономерность: чем чаще слова используются, тем они короче. Применение шаблона "*" в коротких словах приводит к большому количеству ненужных ссылок в списке найденных документов.

Таким образом, актуальным является создание системы полнотекстового поиска, учитывающей описанные выше морфологические особенности.

3. Цель и задачи исследования

Авторами предлагается модель системы полнотекстового поиска по ключевым словам, учитывающая морфологические особенности русского языка, а также алгоритмы индексации и полнотекстового поиска, которые будут использоваться системой.

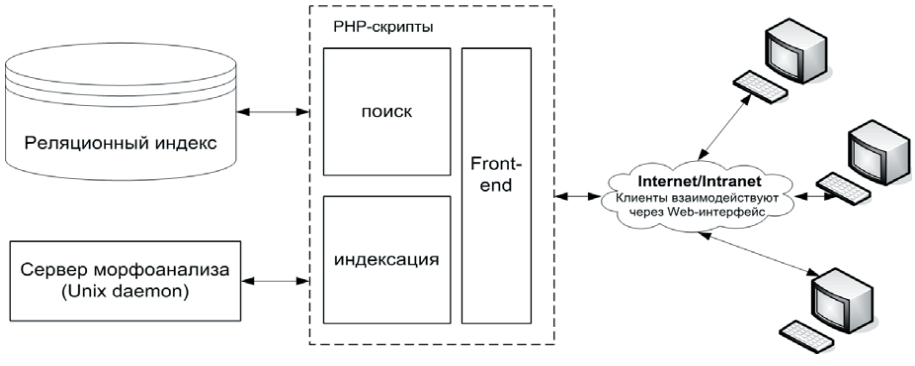

Рис. 1. Общая структурная схема системы

4. Описание системы полнотекстового поиска

Основной функцией разработанной системы является осуществление индексации текстовых документов, чтобы затем можно было быстро и эффективно производить поиск необходимой информации по ключевым словам.

Система учитывает описанные выше морфологические особенности русского языка и реализует поиск информации по индексу, хранящемуся в реляционной базе данных.

Алгоритмы морфологического анализа и синтеза, основанные на базовом словаре, умеют нормализовать слова, то есть находить их начальную форму, а также строить гипотезы для слов, не содержащихся в базовом словаре. В качестве морфоанализатора используется Диалинг [8]. Система полнотекстового индексирования позволяет создавать компактный индекс и быстро осуществлять поиск. Выбор морфоанализатора был обусловлен его возможностью проводить нормализацию со скоростью, большей, чем у других рассмотренных морфологических анализаторов, а также возможностью выполнять выдвижение гипотез в случае, если исходная словоформа не была найдена в словаре. Кроме того данный морфоанализатор распространяется под лицензией LGPL с открытым исходным кодом, которая позволяет бесплатно использовать его при разработке сторонних программных продуктов.

Русский морфологический словарь Диалинг базируется на грамматическом словаре А. А. Зализняка [9]. Включает на данный момент 161 тыс. лемм.

При лемматизации для каждого слова входного текста выдается множество морфологических интерпретаций следующего вида:

- лемма (всегда пишется большими буквами);
- морфологическая часть речи;
- набор общих граммем (которые относятся ко всем словоформам парадигмы слова);
- множество наборов граммем [10].

На рис. 1 изображена общая структура разработанной системы поиска. Основная часть, содержащая в себе бизнес-логику и представление, выполнена на PHP. В качестве реляционной базы данных полнотекстового индекса используется СУБД MySQL. Функции морфоанализа выполняет «демон» normalizer, представляющий собой tcp-сервер и включающий морфологический анализатор Диалинг.

Ядро системы составляют скрипты индексации и поиска, взаимодействующие с полнотекстовым индексом в БД и сервером морфоанализа. Такая двухуровневая модель позволяет более гибко производить развертывание приложения. Например, для увеличения производительности СУБД, морфоанализатор и скрипты поиска/индексации могут быть установлены на отдельных серверах.

На рис. 2 изображена схема реляционный базы данных, которая используется в качестве полнотекстового индекса.

Таблица Documents хранит сведения о проиндексированных документах (имя документа, имя файла на диске). Таблица windex представляет собственно индекс и устанавливает связь между документами и содержащимися в них словами, при этом подсчитывается число вхождений слова в документ. Таблица positions содержит позиции слова в исходном тексте. В таблице stopwords хранятся так называемые стоп-слова, не учитывающиеся при поиске и индексации.

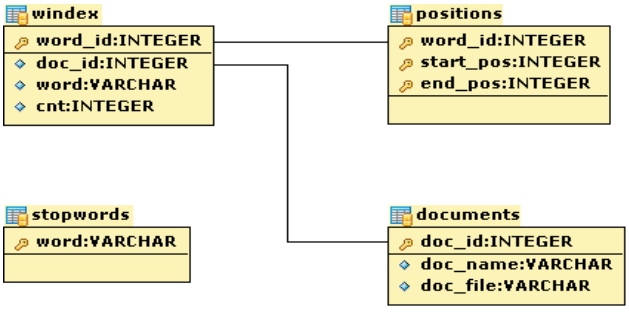

Рис. 2. Схема индексной базы данных

Перед проведением поиска документ должен быть проиндексирован системой.

Индексация – процесс, позволяющий представить документ в форме удобной для поиска и обеспечивающей минимальное время нахождения нужной информации. Индексация включает несколько этапов, среди

которых – этап нормализация словоформ. Таким образом, на выходе процедуры индексации документа – конкорданс (индекс, позволяющий искать с учетом морфологии).

Подробнее, процедура индексации включает следующие этапы:

1) перекодировка при необходимости документа в KOI8-R;

2) разбиение документа на отдельные слова с запоминанием позиции каждого слова в документе;

3) нахождение нормальных форм слов (если у слова есть несколько нормальных слов – то все они учитываются);

4) фильтрация стоп-слов (согласно содержимому таблицы stopwords), а также слов короче 3 символов (в большинстве случаев короткие слова не несут смысловой нагрузки, и ими можно пренебречь при индексации);

5) подсчет частоты появления каждой нормальной формы в документе;

6) запись полученного конкорданса в БД.

Процедура индексации является наиболее требовательной к аппаратной части, т.к. происходит обработка больших массивов текстовых данных. Скорость индексации разработанной системы составляет 300-400 КБ/сек.

Пользователь имеет возможность добавлять новые файлы в индекс при помощи соответствующего интерфейса, позволяющего указать название документа, текстовый файл и его кодировку.

После того, как документ добавлен в индекс, по нему может быть выполнен поиск. Процедура поиска происходит следующим образом:

1) пользователь вводит в качестве поискового запроса ключевые слова или фразу, которые должны присутствовать в искомом документе. Между словами в поисковом запросе неявно ставится логическая операция «и», т.е. «все слова»;

2) система считывает запрос, разбивает его на отдельные слова, при этом используется максимум до 10 ключевых слов из запроса;

3) происходит нормализация словоформ запроса и фильтрация дубликатов, при этом если у слова есть несколько нормальных форм, то будет использоваться только первая. Т. к. при индексации учитывались все нормальные формы, то при поиске достаточно ограничиться лишь одной;

4) фильтрация через стоп-лист слов и отбрасывание слов, короче трех символов;

5) система формирует запрос к реляционной базе данных, хранящей конкорданс, при этом в результаты поиска включается первая тысяча документов, удовлетворяющих условию. Все документы упорядочены в порядке убывания релевантности. В качестве критерия оценки релевантности используется суммарное количество всех найденных слов в тексте документа;

6) выполняется процедура реферирования для результатов поиска с целью включения в список документов, удовлетворяющих условию поиска, текстовой информации, которая поможет пользователю оценить насколько адекватен данный текстовый файл его запросу. Алгоритм реферирования выбирает из документа блок текста фиксированного размера, в котором встречается наибольшее число поисковых терминов, все ключевые термины подсвечиваются. Алгоритмы подсветки и реферирования используют данные из таблицы positions об исходном расположении слов в тексте;

7) полученный список отображается пользователю;

8) пользователь может при помощи щелчка на названии документа перейти к его тексту с подсвеченными поисковыми терминами.

Т. к. индекс хранится в реляционной БД, то задачи низкоуровневого поиска в индексе решает СУБД, при этом важно правильно оптимизировать SQL-запросы к ней, чтобы обеспечить минимальное время поиска. В качестве СУБД была выбрана СУБД MySQL, как бесплатная и одна из самых быстрых.

При поиске PHP-скрипт формирует запрос к БД. Например, если пользователь ввел в строке поиска слова «word1 word2», то SQL-запрос будет иметь вид:

SELECT doc_id, SUM(cnt) AS cnt_row FROM windex WHERE word LIKE 'word1' OR word LIKE 'word2' GROUP BY doc_id HAVING COUNT(word) = 2 ORDER BY cnt_row DESC LIMIT 0,10

Этот запрос выполняет выборку всех документов, в которых встречаются эти 2 слова (word1 И word2). Выборка осуществляется постранично при помощи ключевого слова LIMIT.

Кроме того необходимо создать все требуемые индексы для таблицы, чтобы обеспечить максимальную скорость поиска. Так, на текстовом поле word должен быть создан индекс. Полностью определение таблицы windex приведено ниже:

```
CREATE TABLE `windex` (
`word_id` int(11) unsigned NOT NULL auto_increment,
`doc_id` int(11) unsigned NOT NULL default '0',
`word` varchar(50) NOT NULL default '',
`cnt` int(11) unsigned NOT NULL default '0',
PRIMARY KEY (`word_id`),
KEY `word` (`word`,`cnt`,`doc_id`)
);
```

Система обладает удобным web-интерфейсом пользователя и благодаря клиент-серверной парадигме web-протокола позволяет сразу нескольким пользователям осуществлять поиск и просматривать результаты.

5. Оценка эффективности результатов исследования

Для оценки качества поиска, осуществляемого системой полнотекстового поиска, по формулам (1) –(3) соответственно, рассчитывались такие показатели: *Recall* – мера полноты, *Precision* – мера точности, *Error* – ошибка [11]:

$$\mathrm{Recall} = \frac{a}{a+b}, \quad (1)$$

$$\mathrm{Precision} = \frac{a}{a+c}, \quad (2)$$

$$\mathrm{Error} = \frac{b+c}{(a+b+c+d)}, \quad (3)$$

где a – количество выданных релевантных документов; b – количество выданных нерелевантных до-

кументов; *c* – количество не выданных релевантных документов; *d* – количество не выданных и нерелевантных документов.

Тестирование системы проводилось на коллекции русскоязычных полнотекстовых документов различной тематики. Было исследовано более 1000 документов, общий объем – 82,15 МБ текста (более 5 млн. слов).

Для нашей системы были получены следующие результаты: *Recall* = 0,873; *Precision* = 0,913; *Error* = 0,011. В идеальной системе показатели *Recall* и *Precision* равны 1, *Error* – 0.

6. Выводы

Разработанная система полнотекстового поиска по ключевым словам обладает рядом достоинств. Во-первых, учет морфологии русского языка позволяет производить поиск с оптимальным соотношением избирательности и чувствительности. Во-вторых, система способна работать с достаточно большими индексами. В-третьих, система позволяет сразу нескольким пользователям осуществлять поиск и просматривать результаты.

Тестирование системы на массиве объемом более 5 млн. слов, показало эффективность ее работы.

Литература

1. Ландэ, Д. В. Основы интеграции информационных потоков [Текст]: монография / Д. В. Ландэ. – К.: Инжиниринг, 2006. – 240 с.
2. Ландэ, Д. В. Основы концепции глубинного анализа текстов (Text Mining) [Электронный ресурс] / Д. В. Ландэ. – Режим доступа : http://download.yandex.ru/class/lande/lande-11-tmining.ppt.
3. Бондаренко, М. Ф. О прикладных задачах машинной лингвистики, решаемых подсчетом частот слов и выражений [Текст] / М. Ф. Бондаренко, В. И. Рублинецкий, В. А. Чикина // Проблемы бионики. – Х. : ХИРЭ. – 1999. – Вып. 50. – С. 5-15.
4. Алисейко, З. А. Автоматизированное индексирование полнотекстовых документов ключевыми словами [Текст] / З. А. Алисейко, О. В. Канищева // Вестник Херсонского национального технического университета. – Херсон : ХНТУ. – 2007. – № 4(27). – С. 269-272.
5. Алисейко, З. А. Исследование проблем ранжирования и релевантности полнотекстовых документов в информационном поиске [Текст] / З. А. Алисейко, Н. В. Шаронова // Вестник Херсонского национального технического университета. – Херсон : ХНТУ. – 2006. – № 1(24). – С. 232-236.
6. Хайрова, Н. Ф. Автоматизированные информационные системы: задачи обработки информации [Текст] / Н. Ф. Хайрова, Н. В. Шаронова. – Х.: ХГУ «НУА», 2002. – 120 с.
7. Кочуева, З. А. Моделирование процедур систематизации и классификации информационных объектов методом компараторной идентификации [Текст] / Н. В. Борисова, З. А. Кочуева, Н. В. Шаронова, Н.Ф. Хайрова // Вестник Херсонского национального технического университета. – Херсон : ХНТУ. – 2012. – № 1(44). – С. 91-95.
8. Автоматизированная обработка текста [Электронный ресурс]. – Режим доступа : http://www.aot.ru/.
9. Зализняк, А. А. Грамматический словарь русского языка: Словоизменение [Текст] / А. А. Зализняк. – М.: Рус. яз., 1980. – 880 с.
10. Бондаренко, М. Ф. Автоматическая обработка информации на естественном языке: Учебное пособие [Текст] / М. Ф. Бондаренко, А. Ф. Осыка. – К.: УМК ВО, 1991. – 144 с.
11. Маннинг, К. Введение в информационный поиск [Текст] / К. Маннинг, П. Рагхаван, Х. Шютце. – М.: Вильямс, 2011. – 528 с.