УДК 004.021:004.738.52

МЕТОДЫ И АЛГОРИТМЫ ИНФОРМАЦИОННОГО ПОИСКА НА НЕТОЧНОЕ СООТВЕТСТВИЕ

А.П. ШОРКИН

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь

Поступила в редакцию 11 октября 2010

Приводится сравнительный анализ существующих методов и алгоритмов информационного поиска на неточное соответствие.

Ключевые слова: информационный поиск, неточное соответствие, нечеткий поиск по сходству, расстояние редактирования, методы, алгоритмы.

Введение

В настоящее время алгоритмы информационного поиска на неточное соответствие получили широкое распространение в поисковых системах. Большинство электронных библиотек – это коллекция документов, снабженных инвертированным индексом для быстрого доступа. Центральным звеном поисковых модулей является словарный поиск. Ошибки и искажения могут быть как во вновь добавляемых в систему документах, так и в запросах пользователей, поэтому задача эффективного нечеткого словарного поиска возникает как на этапе создания документа, так и на этапе поиска в уже проиндексированной коллекции [1]. Ниже приводится сравнительный анализ реализованных методов и алгоритмов нечеткого словарного поиска.

Теоретический анализ

Функция похожести строк – это краеугольный камень нечеткого словарного поиска. Выбор подходящей функции похожести влияет не только на качество выборки и скорость поиска, но также и на сложность реализации индекса. Хорошая функция близости слов учитывает различные типы искажений, а в идеале и похожесть звучания слов.

Одна из первых предложенных мер близости слов – это расстояние Левенштейна, равное минимальному числу элементарных операций редактирования, необходимых для преобразования одной строки в другую. Определенное таким образом расстояние редактирования (k) может быть вычислено методом динамического программирования. Алгоритм имеет сложность O(MN), где M и N – длины сравниваемых строк, а для нахождения значения расстояния требуется вычислить MN элементов так называемой матрицы динамического программирования.

Под нечетким поиском по сходству, который является одним из наилучших способов повышения точности поисковых результатов, подразумевается нахождение всех слов, для которых расстояние k до поискового шаблона не превышает заданную величину. Посредством комбинирования различных методов сэмплирования и индексирования происходит образование многочисленных алгоритмов нечеткого поиска по сходству [1]. Поскольку даже наиболее распространенные из них имеют множество модификаций, ниже будут описаны и проанализированы исключительно реализованные методы информационного поиска на неточное соответствие.

Методика

Для реализации поиска на неточное соответствие используются следующие методы:
- последовательный перебор;
- расширение поисковой выборки;
- *n*-граммная индексация;
- выработка хеш-функций;
- trie-деревья (лучи);
- сети Хемминга.

Метод последовательного перебора реализовывается путем последовательного считывания строк и их сравнением непосредственно с поисковым образцом. Главными достоинствами метода являются простота реализации и относительно высокая скорость работы [2]. Объясняется это тем, что при выполнении последовательного считывания файлов большого размера (при условии невысокой фрагментации носителя) достигается пиковая скорость чтения. Помимо этого, данный метод позволяет выполнять многофункциональную поисковую процедуру по подстрокам и регулярным выражениям без ввода значительных ограничений.

Метод расширения выборки заключается в следующем. Предположим, что слово u отличается от слова *v* ровно на одну операцию редактирования. Если построить множество всех слов, получающихся из *u* в результате одной вставки, замены или удаления символа, то полученное множество будет содержать *v*. Это свойство можно использовать для сведения нечеткого поиска к точной выборке. Преимущество алгоритма заключается в том, что время поиска практически не увеличивается с ростом числа записей в словаре. Для индекса, загруженного в память, и максимально допустимого расстояния редактирования равного единице, – это самый быстрый алгоритм. Основной недостаток в том, что он практически не применим для поиска с максимально допустимым расстоянием *k,* большим единицы [1].

Метод *n*-грамм. Словарная *n*-граммная индексация основана на следующем свойстве: если слово *u* получается из слова *w* в результате не более чем *k* элементарных операций редактирования (за исключением перестановок символов), то при любом представлении *u* в виде конкатенации из *k*+1-ой строки, одна из строк такого представления будет точной подстрокой *w*. Таким образом, задача поиска сводится к задаче выборки всех слов, содержащих заданную подстроку [3]. Для решения этой задачи удобно использовать инвертирование относительно набора *n*-грамм слова. Скорость поиска линейно зависит от числа проиндексированных слов, поэтому главный недостаток метода кроется в больших размерах файла.

Метод хеширования реализуется разработкой хеш-функции, определяющей основные характеристики слова; при этом сама функция обладает устойчивостью к возможным искажениям. Хеш-функция делает разбивку слова на определенные группы и выполняет поисковую процедуру последовательно по каждой группе, которая вычисляется на базе шаблона поиска. Примером хеш-функции является программная функция soindex, реализованная в СУБД MS SQL Server, Oracle и др. Данный метод soindex реализует «частичный» поиск, поскольку хеш-функция обладает устойчивостью лишь на узком классе искажений и достаточно сложно предсказать, какие изменения будут получены в результате вставки случайных символов [4].

Строковые trie-деревья. Суть этого метода заключается в следующем. На самом верхнем уровне группируются слова с одинаковой первой буквой. Эти слова составляют отдельное поддерево. Соответственно, на втором уровне дерева слова группируются по значению второй буквы и т. д. Для узлов, имеющих ровно одного потомка, применяется алгоритм сжатия путей – такие узлы объединяются в один. В процессе спуска по дереву происходит вычисление матрицы динамического программирования. Для каждой добавляемой при спуске букве необходимо вычислить новый столбец матрицы. Если в процессе поиска в последнем столбце нет элементов, меньших максимально допустимого расстояния редактирования, то поддеревья текущего узла можно исключить из дальнейшего поиска. На практике алгоритм trie-деревьев реализуется для поиска по подстрокам, но его также можно использовать, и весьма эффективно, для осуществления поиска по сходству [4].

Сети Хемминга. Для нечеткого поиска могут быть использованы алгоритмы и коды Хемминга, которые успешно применяются при кодировании и декодировании информации. Реализация метода происходит следующим образом. На вход поисковой системы подается ис-

комое слово *v* и текстовый файл со списком слов *V*, в котором будет осуществляться поиск. На выходе получается номер n (*w*, *V*) слова *w* в списке *V*, которое наиболее близко к исходному слову *v*.

Входное слово из букв русского алфавита преобразуется в слово в алфавите 01, которое затем подается на вход нейронной сети, т.е. каждой букве ставится в соответствие слово из символов 0 и 1 длины 5. Кодирование строится таким образом, чтобы стоящие рядом на компьютерной клавиатуре символы имели близкие по Хеммингу коды (например, «ф» – 00001, «ы» – 00011), что обеспечивает наиболее эффективное исправление опечаток [5].

В этой реализации размерность распределительного слоя фиксирована и составляет 125 нейронов. Размерность скрытого слоя определяется количеством слов в списке, в котором осуществляется поиск. Размерность выходного слоя равна размерности скрытого слоя нейронной сети. Реализация сети показала высокие результаты по обобщению данных и нахождению всех тестовых слов. Стоит отметить, что система хорошо исправляет опечатки, но не пропуски и лишние символы, при которых Хеммингово расстояние является слишком большим.

Заключение

При сравнительном анализе основных методов поиска были обнаружены их ключевые достоинства и недостатки. Плюсом более простых алгоритмов является высокая скорость работы, а минусом – большой размер поискового индекса. Из наиболее эффективных алгоритмов следует отметить алгоритмы *n*-грамм и trie-деревьев, которые обеспечивают хорошее соотношение между размером индекса и скоростью поиска. Отдельного внимания заслуживает алгоритм Хемминга, на базе которого возможна реализация эффективных моделей нечеткого поиска по сходству. Однако перечисленные методы имеют свои недостатки, поэтому проблема поиска с учетом более общей функции близости остается открытой.

METHODS AND ALGORITHMS OF INDORMATION SEARCHING USING INEXACT MATCHING

A.P. SHORKIN

Abstract

Analysis of existing approaches solving informational text searching using inexact matching was made. Problem area of the low quality level in standard search process using keywords was decrypted. Algorithms of improving search results taking into account language and context relations between query terms were dedicated.

Литература

1. *Бойцов Л.М.* // Труды 6-ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» 2004.
2. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М., 2001.
3. *Стефан А.* Анализ строк. Пер. с англ. М.С. Галкиной под ред. под ред. П.Н. Дубнера.
4. *Бойцов Л.М.* // Прикладная математика и информатика. 2005. №7.
5. *Головко В.А.* Нейронные сети: обучение, организация и применение. М., 2007.