УДК 681.3

ПОИСКОВЫЕ МЕТОДЫ НА ОСНОВЕ TREE-ДЕРЕВЬЕВ

© 2016 М. А. Демихов

Воронежский институт высоких технологий

В работе описываются методы, базирующиеся на так называемых tree-деревьях. При этом в поиске ориентируются не на точную, а релевантную информацию. Дана классификация tree-деревьев и времен их поиска.

Ключевые слова: поисковый метод, структура данных, tree-дерево.

Поисковые методы на основе tree-деревьев. В последнее время возросла тенденция к обработке больших объемов текстовой информации, в первую очередь, это связано с развитием сети Интернет, которое привело к появлению огромного количества текстовой информации, которые представляются как обычные тексты, HTML и XML документы, сообщения электронной почты. В современных системах хранения информации основной задачей являются надежность хранения данных и высокая скорость методов поиска, причем основная нагрузка на поисковые системы идет в виде поиска не точной, а релевантной информации, где релевантность определяется степенью смысловой близости к изначальному запросу.

Время поиска в структурах, представленных в виде списков, по текстовым данным составляет O (logn), при этом списки ключевых терминов должны быть отсортированы, так как именно при удовлетворении такого условия возможен бинарный поиск за логарифмическое время. Отсортированные списки имеют еще один недостаток: их сложно модифицировать (удалять / вставлять ключевые термины) и количество операций затрачивается на это не меньше, чем O(n). Следует отметить, что существуют структуры данных, которые не обладают выше изложенными недостатками – это древовидные структуры данных. В данной главе будут проанализированы методы поиска полнотекстовой информации, представленной в древовидной структуре. Виды древовидных (tree) структур Суффиксные деревья (suffix tree) Суффиксное дерево (suffix tree, PAT tree, position tree) – это префиксное дерево (trie), содержащее все суффиксы заданного текста (ключи) и их начальные позиции в тексте.

Демихов Михаил Александрович – ВИВТ АНОО ВО, магистр, e-mail:msqwfrdemikhovamm@yandex.ru.

Суффиксное дерево T для строки $S=s_i,\ldots, s_n$, является структурой данных, которая имеет следующие свойства: это дерева, имеющее корневой узел, в котором точно n листьев, которые пронумерованы от 1 до n. В каждом внутреннем некорневом узле есть по крайней мере 2 ответвления. Для каждого ядра дерева, есть пометка непустой подстроки строки S, нет более чем 2 ребер, которые идут из одного узла, чтобы они были помечены строками, которые начинаются с одних и тех же символов. Для любого из листов i в дереве T, на основе соединения всех меток, относящихся к ребрам по путям от корней до листа i образуется подстрока S_i = si, ... , sn , строки S.

Данный алгоритм был впервые введен Вайнером, который Дональд Кнут впоследствии охарактеризовал как «Алгоритм года 1973». В 1995 году был разработан первый алгоритм построения суффиксного дерева для заданной строки S за линейное время, сейчас известный как «Алгоритм Укконена». Большое число различных задач можно решать на основе суффиксных деревьев: это касается проблем, связанных с поиском наибольших общих подстрок, комплиментарных пар для последовательностей ДНК, также есть задачи сжатия информации, проведением нечеткого поиска и процессов кластеризации.

Практическое применение:

– поиск подстроки в строке O(м), где м – длина подстроки (но требуется O(n) времени для построения суффиксного дерева для строки);

– поиск наибольшей повторяющейся подстроки;

– проведение поиска по наибольшей общей подстроке;

– проведение поиска по наибольшей подстроке палиндрома;

– алгоритм LZW сжатия информации (они помогают найти повторяющиеся данные). В качестве примера на рисунке изображено суффиксное дерево по тексту BANANA. Суффиксы: «A$», «NA$», «ANA$», «NANA$», «ANANA$», «BANANA$».

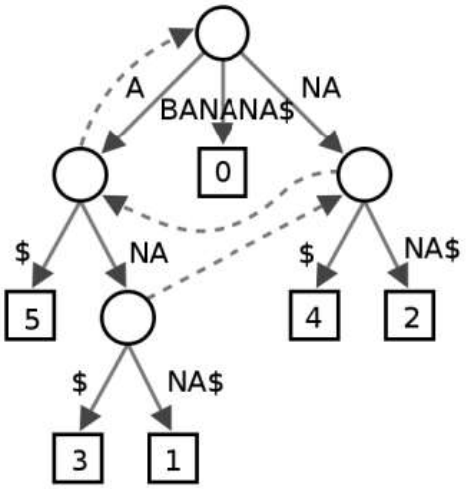

Рис. 1. Суффиксное дерево

Префиксные деревья (trie)

Префиксное дерево (Trie, prefix tree, digital tree, radix tree) – это структура данных для реализации словаря (ассоциативного массива), ключами в котором являются строки. Особенностью данной структуры заключается в том, что ключи не хранятся в узлах такого дерева.

Практическое применение:

- предиктивный ввод текста (predictive text) – поиск возможных завершений слов;
- автозавершение (Autocomplete)в текстовых редакторах и IDE;
- проверка правописания (spellcheck);
- автоматическая расстановка переносов слов (hyphenation);
- Squid Caching Proxy Server.

На рисунке 2 приведен пример префиксного дерева для ключей: "A", "to", "tea", "ted", "ten", "i", "in", and "inn".

Тернарное дерево – один из видов префиксного дерева (trie), где узлы располагаются в виде бинарного дерева.

Можно указать метод поиска, связанный с набором строк, основанный на тернарных деревьях поиска (ternary search trees). Первый раз эти деревья были упомянуты еще в 1964 году, но лишь в последнее время идеи, которые лежали в основе таких структур данных были практическим образом развиты. Например, требуется провести поиск для следующих двенадцати двухбуквенных строк: "as" "at" "be" "by" "he" "in" "is" "it" "of" "on" "or" "to".

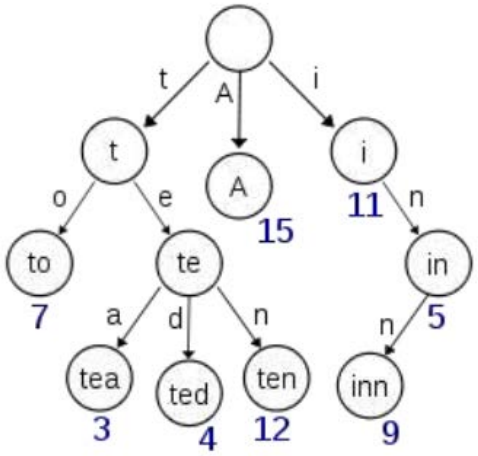

Рис. 2. Префисное дерево

Тернарное дерево будет выглядеть следующим образом:

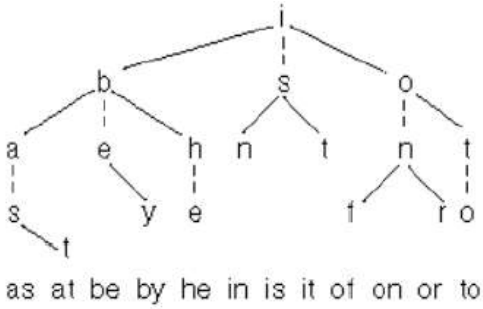

Рис. 3. Тернарое дерево

По сравнению с хеш-таблицами тернарные деревья позволяют очень быстро вернуть отрицательный результат по поиска для шаблонов, заведомо не содержащихся в исходном документе.

Важно указать, что тернарные деревья дают возможности также для эффективного осуществления поиска по сходству, и уже давно функционирует система, связанная с автоматическим распознаванием текстов фирмы Bell Labs (OCR System), в ней для проведения поиска терминов для словаря английского языка, имеющего объем около 34,000 символов первый раз были применены тернарные деревья поиска.

Их можно использовать для решения многих проблем, в которых необходимо хранить большое количество строк и извлекать в произвольном порядке.

Поиск ключей в худшем случае будет выполняться за O(м) в префиксных деревьях, т. к. время поиска не зависит от количества n элементов в словаре, а только от длины ключа (м). В бинарных деревьях сравне-

ние ключей происходит за O(n), где n количество элементов в дереве, т.к. поиск зависит от глубины дерева. Таким образом, в худшем случае поиск в бинарном дереве занимает O(n).

Префиксные деревья могу быть медленнее, например, чем хэш-таблицы в случае поиска данных, особенно если данные хранятся непосредственно на жестком диске или другом устройстве хранения время произвольного доступа является высокой по сравнению с основной памятью. Суффиксные деревья обеспечивают поиск за (2k) операций сравнения, где к является числом всех вхождений требуемой подстроки в исходную строку.

Также требуется O(n) линейного времени для построения суффиксного дерева для строки.

Время выполнения поиска в тернарных деревьях значительно меняется от входных данных, например, поиск будет осуществляться лучше, если будет даваться несколько подобных строк, особенно когда эти строки имеют общий префикс. Кроме того, поиск в тернарных деревьях является более эффективным при хранении большого количества относительно коротких строк. Поиск по ключам в тернаных деревьях происходит за O(n+1), худшее время выполнения O(n).

ЛИТЕРАТУРА

1. Карахтанов Д. С. Использование алгоритмов нечеткого поиска при решении задач обработки массивов данных в интересах кредитных организаций / Д. С. Карахтанов // Аудит и финансовый анализ. – 2010. – № 2. – URL: www.auditfin.com/2010/2/toc.asp.

2. Бойцов Л. М. Классификация и экспериментальное исследование современных алгоритмов нечеткого словарного поиска / Л. М. Бойцов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды VI Всеросс. научн. конф.(RCDL'2004). – Пущино, Россия, 2004. URL: http://rcdl.ru/doc/2004/paper27.pdf.

3. Карахтанов Д. С. Использование алгоритмов нечеткого поиска при решении задачи устранения дубликатов в массивах данных/ Д. С. Карахтанов // Молодой ученый. – 2010. – Т. 1. – № 8 (19). – С. 150-155.

4. Потапов Е. Н. Нечеткие множества в хранилище данных. / Е. Н. Потапов // 2011. URL: http://разработкахд.рф/blog/? p=346.

5. Рыжов А. П. Модели поиска информации в нечеткой среде / А. П. Рыжов. – М.: Изд-во ЦПИ при ММФ МГУ, 2004. – 96 с.

6. Преображенский Ю. П. Некоторые аспекты информатизации образовательных учреждений и развития медиакомпетентности преподавателей и руководителей / Ю. П. Преображенский, Н. С. Преображенская, И. Я. Львович // Вестник Воронежского государственного технического университета. – 2013. – Т. 9. – № 5-2. – С. 134-136.

7. Преображенский Ю. П. Разработка лингвистических средств интеллектуальной поддержки принятия медицинских решений в клинической практике на основе имитационно-семантического моделирования / Ю. П. Преображенский, Н. С. Преображенская, В. В. Ермолова // Information Technology Applications. – 2013. – № 4. – С. 96-114.

8. Преображенский Ю. П. Сравнительный анализ алгоритмов поиска текстовых фрагментов / Ю. П. Преображенский, А. С. Ермаченко // Вестник Воронежского института высоких технологий. – 2010. – № 7. – С. 76-78.

9. Фомина Ю. А. Принципы индексации информации в поисковых системах / Ю. А. Фомина, Ю. П. Преображенский // Вестник Воронежского института высоких технологий. – 2010. – № 7. – С. 98-100.

10. Паневин Р. Ю. Реализация транслятора имитационно-семантического моделирования / Р. Ю. Паневин, Ю. П. Преображенский // Вестник Воронежского института высоких технологий. – 2009. – № 5. – С. 057-060.

11. Зазулин А. В. Особенности построения семантических моделей предметной области/ А. В. Зазулин, Ю. П. Преображенский // Вестник Воронежского института высоких технологий. – 2008. – № 3. – С. 026-028.

12. Иванов М. С. Разработка алгоритма отсечения деревьев / М. С. Иванов, Ю. П. Преображенский // Вестник Воронежского института высоких технологий. – 2008. – № 3. – С. 31-32.

13. Паневин Р. Ю. Структурные и функцинальные требования к программному комплексу представления знаний / Р. Ю. Паневин, Ю. П. Преображенский // Вестник Воронежского института высоких технологий. – 2008. – № 3. – С. 61-64.

14. Максимова А. А. Анализ методов обработки медицинских данных / А. А. Максимова // Моделирование, оптимизация и ин-

формационные технологии. – 2016. – № 2. – С. 5.

15. Мэн Ц. Анализ методов классификации информации в интернете при решении задач информационного поиска / Ц. Мэн // Моделирование, оптимизация и информационные технологии. – 2016. – № 2. – С. 19.

16. Чопоров О. Н. Оптимизация управления функционированием медицинских систем различного уровня / О. Н. Чопоров, И. Я. Львович, К. А. Разинкин, А. А. Рындин // Системы управления и информационные технологии. – 2013. – Т. 53. – № 3. – С. 100-104.

THE SEARCH-BASED METHODS OF TREE-TREES

© 2016 M. A. Demihov

Voronezh institute of high technologies

The paper describes methods based on the so-called tree-trees. In the search focus is not on accurate and relevant information. A classification tree is a tree and the time of their search.

Keywords: search method, data structure, tree.