УДК 621.396

ПРОБЛЕМЫ ПОИСКА ТЕКСТОВОЙ ИНФОРМАЦИИ В БОЛЬШИХ ОБЪЕМАХ ДАННЫХ

© 2019 А. В. Шапаев, Д. А. Юдаков, А. А. Часовской

Воронежский институт высоких технологий (г. Воронеж, Россия)
Московский гуманитарно-экономический университет (г. Москва, Россия)

В статье обсуждаются задачи, связанные с поиском информации в больших информационных массивах. Описаны характеристики созданного программного продукта для поиска информации.

Ключевые слова: текстовая информация, данные, поиск, алгоритм.

В настоящее время наблюдается рост объемов информации, которые как хранятся в различных базах данных, так и передаются по каналам связи.

Представляет интерес провести анализ основных особенностей, связанных с поиском информации в больших объемах данных.

Предположим, что рассматриваем некоторую последовательность символов. Возникает задача, связанная с поиском символов, которые будут идти друг за другом. Говоря иными словами, в большом блоке требуется найти подблок.

Этот подблок может находиться на конечном числе позиций. Их число связано с общим размером большого блока и размером подблока.

По каждой такой позиции есть возможности для того, чтобы была осуществлена проверка за счет проверки нескольких символов. Путем последовательного перебора подряд мы, просматривая в определенном направлении символы, сравнивая их с символами в подблоке имеем шансы обнаружить полный подблок.

Подобные алгоритмы могут быть описаны с точки зрения метода конечных автоматов.

Любая последовательность, составленная из символов, будет являться словом. Оно формируется на базе определенного фиксированного конечного множества [1, 2].

Его рассматривают в виде алфавита. Буквы представлены элементами множества. Сколько подмножеств, столько можно сформировать и слов. Просматривать их или читать мы можем, например, слева направо.

Задача может быть описана и на основе индуктивных функций. Тогда, функция, если принимает значение «истина», то слово будет найдено, и если «ложь», то слово не найдено.

Можно ли сделать замену анализируемого слова на любое слово? Дело в том, что внутри образцов можно встретить повторяющиеся буквы. Поэтому ответ будет – нет.

Не сложно в случае, когда есть соответствующий образец, сформировать программный продукт, в рамках которого будет возможен поиск по такому образцу. Но, для более общего случая [3, 4] представляет интерес создания программы, на основе которой бы произвольный образец был бы найден для произвольного слова.

Осуществить это можно при помощи двух шагов. В первую очередь, таблицу переходов конечных автоматов формируют на базе образца. После этого происходит чтение входного слова. Идет преобразование состояния исходя из такой таблицы.

Чтобы просто найти подблок может быть применен алгоритм Кнута-Моррисa-Пратта, который является более простым.

В тех случаях, если вначале будет на вход поступать блок, а потом подблок, предназначенный для поиска, целесообразно опираться на метод суффиксных деревьев.

Можно продемонстрировать, что процесс формирования сжатого суффиксного дерева может быть исполнен в течение времени O (t^2) при использовании O (t) памяти.

В суффиксное дерево будут постепенно добавляться суффиксы. Проведение провер-

Шапаев Александр Викторович – Воронежский институт высоких технологий, аспирант, Shapaevv3456@yandex.ru.
Юдаков Дмитрий Артурович – Московский гуманитарно-экономический университет, специалист, gtpyudak391@ymail.com.
Часовской Алексей Алексеевич – Воронежский институт высоких технологий, аспирант.

ки принадлежности осуществляется также, как и процесс добавления следующих суффиксов. Слова читаются последовательным образом и, таким образом, будет обозначен путь внутри деревьев. Когда за дерево выйдет вносимый суффикс, то это будет соответствовать тому, что требуется осуществить разрез по ребру.

Если это произойдет посередине ребра, то ребро придётся в этом месте разрезать. Тогда после того, как будет обнаружено места ветвления, будет необходимо провести 0(1) операций, чтобы дерево было перестроено.

Более быстрым является алгоритм Мак-Крейта.

Алгорим Рабина исходит из того, что требуется сформировать некоторую функцию, которая определена на некоторой длине. Если есть разница по значениям по образцу и в анализируемом подблоке в блоке, то подблок не найден, нельзя говорить о совпадении. В противном случае, требуется осуществлять уже сравнение по внутренней структуре подблока. Есть определенный выигрыш в подобном подходе, если сравнивать с простым перебором внутренней структуры блока. Это происходит вследствие того, что при движении подблока внутри блока, меняются начало и конец. Тогда на основе подобных данных можно осуществить расчет того, каким образом будут осуществляться изменении в функции.

В рамках алгоритма Бойера-Мура осуществляется чтение только небольшой части блока, для которого осуществляется поиск заданного подблока.

Алгоритмы поиска могут быть реализованы в программных продуктах (ПП). Для того, чтобы начать работу программы требуется, чтобы был сформирован текстовый документ, имеющий большую длину. Это позволит продемонстрировать, насколько эффективен исследуемый алгоритм [5, 6]. Внутри рабочего окна ПП может быть указан любой из трех критериев поиска информации – это по коду в таблице ASCII от 0 – 255, по пути, где расположена папка и по названию папки.

После сканирования программа автоматически выведет тот текстовый документ или строку где находиться заданный код.

Данный программный комплекс реализован на языке программирования – C++. Выбор обусловлен тем, что данный продукт предоставляет программисту большой спектр решений средств доступа к данным [7, 8], простой реализации требуемых эстетических и эргономических показателей и сведением к минимуму аппаратно-программных требований. Также достигается абсолютная совместимость с линейкой операционных систем Microsoft Windows, ставшей стандартом de facto для современных компьютеров.

Краткая характеристика ПП:

1. Объем ПП: $n_{т.н.к.}$ = 6,7 тыс. исходных команд

2. Группа сложности ПП: 2; дополнительный коэффициент сложности: $К_{сл}$ = 1,12;

3. Степень новизны разрабатываемого ПП: В (ПП является развитием определенного параметрического ряда ПП и разработан на ранее освоенных типах ЭВМ и ОС).

Существующие западные разработки сильно перегружены функциональными возможностями, имеют довольно высокие аппаратно-программные требования (мощный процессор, большой объем оперативной памяти и т. п.), а их относительно высокая цена делает их недоступными для учреждений среднего и малого звена.

В то же время абсолютно четко сформировался круг потенциальных потребителей описанного ПП в лице государственных и коммерческих учреждений [9, 10], оказывающих услуги связи.

Таким образом, возникает необходимость разработки специального пакета прикладных программ, который позволил бы решать задачи серверного доступа, имел минимальные в своем классе аппаратно-программные требования, был совместим с каким-либо пакетом для упрощения ввода исходных данных.

В качестве базового варианта рассмотрим ПП, разработанный ранее описанной системы. В нем использовался стандартный алгоритм Бойера-Мура, отсутствовала, возможность ввести в систему новые элементы и не поддерживался обмен со сторонними базами данных. Таким образом, данная базовая модель не до конца удовлетворяла требованиям, предъявляемым к необходимому ПП.

Рассмотрим недостатки базового варианта:

Невысокая степень интеграции в существующие ОС;

Сложность введения данных в систему;

Отсутствие возможности корректировки БД;

Отсутствие интеграции с существующими пакетами обработки статистической информации.

Все недостатки базового варианта были учтены и реализованы в новой версии ПП.

ЛИТЕРАТУРА

1. Горбенко, О. Н. Характеристики информационных процессов в образовательной среде / О. Н. Горбенко, В. Н. острова // Моделирование, оптимизация и информационные технологии. – 2015. – № 1 (8). – С. 17.

2. Мэн Ц. Анализ методов классификации информации в интернете при решении задач информационного поиска / Ц. Мэн // Моделирование, оптимизация и информационные технологии. – 2016. – № 2 (13). – С. 19.

3. Преображенский, А. П. Анализ информационных процессов в современном образовании / А. П. Преображенский, Е. И. Коденцев // Вестник Воронежского государственного технического университета. – 2013. – Т. 9. – № 5-2. – С. 98-101.

4. Преображенский А. П. Построение многокритериальной модели работы предприятия / А. П. Преображенский, О. Н. Чопоров // Наука Красноярья. – 2017. – Т. 6. – № 3-4. – С. 183-188.

5. Львович, И. Я. О проблемах подготовки инженерных кадров / И. Я. Львович, А. П. Преображенский // Вестник Воронежского государственного технического университета. – 2014. – Т. 10. – № 5-2. – С. 157-160.

6. Преображенский, А. П. Особенности использования информационных технологий при подготовке современных специалистов / А. П. Преображенский, О. Н. Чопоров // В мире научных открытий. – 2015. – № 9-2 (69). – С. 670-675.

7. Львович, И. Я. Снижение количества ошибок распознавания сканированных рукописных текстов / И. Я. Львович, Я. Е. Львович, А. А. Мозговой, А. П. Преображенский, О. Н. Чопоров // Цифровая обработка сигналов. – 2016. – № 4. – С. 43-47.

8. Кострова, В. Н. Оптимизация распределения ресурсов в рамках комплекса общеобразовательных учреждений / В. Н. Кострова, Я. Е. Львович, О. Н. Мосолов // Вестник Воронежского государственного технического университета. – 2007. – Т. 3. – № 8. С. 174-176.

9. Филипова, В. Н. Использование процессов моделирования и управления в туризме / В. Н. Филипова, Ю. А. Пивоварова // Моделирование, оптимизация и информационные технологии. – 2014. – № 2 (5). – С. 19.

10. Кайдакова, К. В. Проблемы защиты информации в современных электронных документах / К. В. Кайдакова // Успехи современного естествознания. – 2012. – № 6. – С. 107-108.

THE PROBLEM OF FINDING TEXTUAL INFORMATION IN LARGE BLOCKS OF DATA

© 2019 A. V. Shalaev, D. A. Yudakov, A. A. Chasovskoy

Voronezh Institute of High Technologies (Voronezh, Russia)
Moscow University of Humanities and Economics (Moscow, Russia)

The problems associated with the search for information in large information arrays are discussed. The characteristics of the created software product for information search are described.

Key words: text information, data, search, algorithm.