Индексирование русских текстов с использованием словаря, представленного на основе разреженной хэш-таблицы

Современные компьютерные программы, анализирующие текст на естественном языке, как правило, используют словари. Цель словарей — помочь распознать встреченную текстовую цепочку и, возможно, обозначить ее уникальным идентификатором.

В технологии, опирающейся на словарный подход к анализу текста, требуется каким-то образом эти словари представлять.

Во многих случаях на представление словаря накладываются дополнительные требования. Необходимо, чтобы он был как можно более компактным, а процедура разбора и идентификации слова занимала как можно меньше времени. Эти требования возникают, например, при анализе большого объема текста и/или в коммерческом программном продукте. Примеры — проверка правописания большого файла; построение пословного индекса для большого текстового массива; процедура, предлагающая варианты написания незнакомого слова.

Опуская частности, анализ (морфологический разбор) слова работает в такой последовательности. Сначала от предполагаемого слова отрезаются все возможные окончания и приставки. Затем каждый гипотетический вариант основы проверяется на наличие в словаре. И, наконец, если такая основа в словаре существует, проверяется соответствие отрезанного окончания той грамматической информации, которая ассоциируется в словаре с данной основой.

Традиционное представление словарей в программах для IBM PC использует блочно-слотовую организацию, хранящую непосредственный текст основы и закодированное описанием морфологии рядом с ним. Размер такого словаря (на 100-120 тыс слов) обычно составляет 600-800 KB, а процедура идентификации основы включает (под MS-DOS непременно, под MS-WINDOWS чаще

всего) чтение одного или нескольких блоков с диска, а затем многократное сопоставление гипотетической основы с основами внутри блока.

Предельно компактное и эффективное решение задачи проверки правописания английских слов было предложено *McIllroy 1978, 1982*. Решение включало процедуру анализа аффиксов (то есть набор правил для отбрасывания префиксов и суффиксов и список исключений из этих правил) и специальное представление текстов основы. Это представление заменяет текст основы на сответствующий ему бит из хэш-таблицы большого размера. Объем памяти, требуемый для представления одной основы, таким образом, сводится до 14 бит. В то же время порождается ошибка, при которой неправильное слово может быть ошибочно принято за правильное. Вероятность возникновения такой ошибки, впрочем, существенно мала, и может быть понижена за счет большего разрежения таблицы.

Идея использовать такое представление в русском грамматическом словаре показалась автору в начале 1993 года весьма заманчивой. Для этого необходимо было решить несколько проблем:

- Русский морфологический анализ требовал организовать для каждой основы хранение ссылки на соответствующую парадигму склонения / спряжения.
- Для задачи индексирования текста необходимо не только проверять факт наличия основы и соответсвия аффикса парадигме. Требуется получить все словарные варианты, соответствующие гипотетической словоформе. Кроме того, каждый вариант разбора (то есть словарную статью) нужно идентифицировать ключом, который и входит в состав пословного индекса. Заметим, что текст нормальной формы (леммы) не может служить таким ключом из-за омонимии.

Результатом специальной трансляции грамматического словаря, полученного автором в ИППИ РАН и представленного в

соответствующем формате, явился словарь на 108 тысяч основ со следующими параметрами:

Занимаемый объем:

- 270 Kb – разреженная хэш-таблица основ вместе с номерами соответствующих парадигм;
- 90 Kb – таблица парадигм;
- 14 Kb – дерево окончаний.
  Скорость разбора русского текста показывает следующая таблица:

Процессор	слов в секунду
286	300–500
386SX	700–900
386DX	900–1300 (до 2000 в 32-битном режиме)
486	2500-4000 (до 6000 в 32-битном режиме)

Окончательный результат был получен после оптимизации всех слагаемых технологии, прежде всего, самой хэш-таблицы и методов доступа к ней:

- Была выбрана подходящая хэш-функция.
- Варьировался размер хэш-таблицы и связанная с ним вероятность ошибки.
- Благодаря тому, что, кроме бита, соответствующего основе, в словаре хранится номер парадигмы, вероятность ошибочных совпадений была резко уменьшена.
- Разреженная хэш-таблица хранится в виде *разностей,* и упаковывается *кодами переменной длины*. Были рассмотрены разные схемы упаковки и выбрана схема, не оптимальная по размеру словаря, но оптимальная по скорости доступа.

- Ссылки на парадигмы, хранящиеся также в виде кодов переменной длины, были перенумерованы в соответствии с частотой встречаемости парадигм в словаре.
- Были рассмотрены разные схемы совместного / отдельного хранения хэш-таблицы и ссылок на парадигмы и выбрана оптимальная.

Оптимизировалась также технология хранения грамматической информации:

- Окончания были перетранслированы в таблицу переходов, имеющую структуру *сильно ветвящегося дерева* и позволяющую за один спуск по нему извлечь все гипотетические варианты разбора.
- Парадигма была представлена в виде упорядоченного массива номеров окончаний, а проверка соответствия окончания парадигме проводилось двоичным поиском.

Компактность словаря и высокая скорость работы с ним окупает некоторую сложность его построения, а отсутствие возможностей синтеза преодолевается в сочетании с исходным грамматическим словарем.

Вокруг хэш-словаря сложилась технология построения информационно-поисковых систем (реализуемая в проекте Яndex), которая основывается на следующих идеях:

- Номер входа в хэш-таблицу используется как уникальный идентификатор и используется при индексировании.
- Разбор поисковых запросов производится синхронно по тому же алгоритму, с использованием того же словаря и с получением того же идентификатора
- Сразу после индексирования обладатель исходного грамматического словаря запускает программу, которая для всех распознанных основ проиндексированного текста строит их нормальные формы (лемматизирует) и связывает разные основы одного слова.
- Результатом *прогона* является закрытая ИПС (например для CD-ROM) с частотным словарем нормальных форм, позволяющем выбрать слово из

имеющихся в тексте, а также компактным хэш-словарем, позволяющем задавать в запросе любые формы этих слов.

Примером ИПС, полученной с помощью хэш-словаря может служить программный продукт "Библейский компьютерный справочник".