Система рубрикации данных «Синдбад»

© В.В. Рыбинкин

ООО «Бюро Интернет Технологий БИТ»
ryb@2bit.ru

Аннотация

В данной работе проведен анализ проблем, возникающих при автоматизированной рубрикации данных, а также особенностей, связанных с рубрикацией слабоструктурированной информации, предложен алгоритм рубрикации текстовой информации. Изложенные подходы практически реализованы при создании специализированного пакета прикладных программ структуризации и каталогизации информации и применимы для рубрикации данных Веб-сайтов, реляционных БД, систем управления документооборотом, текстовых файлов.

1 Введение

Значение автоматической рубрикации данных трудно переоценить. Например, для эффективной работы реляционной СУБД необходимы индексные файлы, а для систем среднего и большого объема желательно выполнять кодирование повторяющихся терминов и, следовательно, иметь словари (классификаторы). Различные классификаторы, рубрикаторы все чаще применяются и для поиска информации, особенно текстовой, неструктурированной [1]. Именно рубрикация обеспечивает возможность доступа к логически связанным неоднородным данным, наглядное представление информации в виде системы каталогов. Проблема рубрикации сложно структурированных данных особенно актуальна в связи с нарастающим влиянием сети Интернет.

Для классификаторов сложной структуры характерно наличие большого количества близких по смыслу рубрик, поэтому для таких рубрик характерно также отклонение результатов автоматической и

ручной классификации документов. Тем не менее, необходимость именно автоматической рубрикации очевидна: конфликт интересов, предвзятость экспертов, отсутствие экспертов в данной области, наконец, чрезвычайно высокая трудоемкость этой операции убедительно показывают чрезвычайную актуальность эффективных процедур автоматической рубрикации.

Обычно под рубрикацией данных понимают иерархическую рубрикацию, т.е. организацию рубрик в виде «дерева». Процедура классификации состоит в индексировании анализируемых документов и рекурсивном обходе дерева рубрик с выполнением на каждом уровне иерархии полнотекстового поиска по терминам, характеризующим ту или иную рубрику [1]. Такая рубрикация автоматически приводит к ряду серьезных проблем. Помимо дублирования данных в различных узлах дерева со всеми вытекающими трудностями их редактирования, добавления или удаления, не говоря уже о катастрофическом возрастании объемов, иерархия неудобна как для размещения информации, так и для ее поиска из-за сложности выбора наиболее адекватного окружения для размещаемого или искомого узла. В самом деле, где искать, например, **Автоспорт** - в **Автомобилях** или в **Спорте**? Подобные вопросы всегда возникают при рубрикации сложно организованной информации, и в рамках иерархической модели данных их удовлетворительное разрешение часто оказывается невозможным.

Мы полагаем, что проблема рубрикации неоднородной неструктурированной распределенной информации может быть решена в рамках сетевой модели данных, поскольку только в ней связи между данными могут быть организованы сколь угодно сложным образом. Целью проведенной нашей компанией разработки, выполненной в рамках НИР и ОКР, и являлось практическое подтверждение этого тезиса.

2 Анализ проблем, возникающих при рубрикации данных

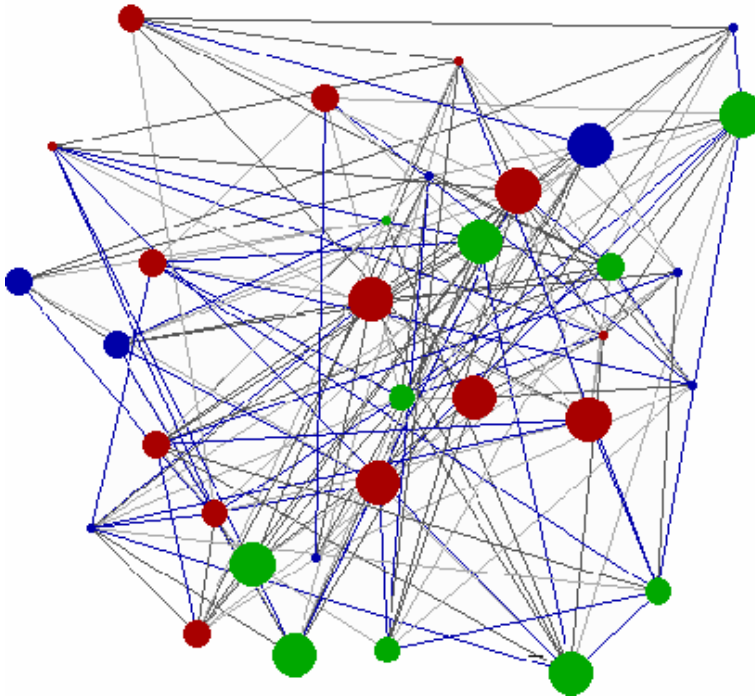
Чтобы говорить о рубрикации в терминах сетевой модели данных, введем несколько определений:

Узел графа рубрик есть именованная структура данных. Под именем узла понимаем его идентификатор (ID).

Ребро графа есть одно из полей узла типа «ссылка», в которой расположен ID узла, связанного с ним.

Тип ребра графа или **измерение** есть именованная характеристика

Рис. 1. Представление массива рубрик в виде графа.



связи между узлами, представленная этим ребром, например: тематическая, географическая, темпоральная, алфавитная и т.д.
Количество измерений в узле назовем **размерностью узла**, общее количество измерений во всех узлах графа – **размерностью графа**. Граф называется **многомерным**, если его ребра расположены в разных измерениях, т.е. размерность графа больше единицы.

Набор рубрик, по которым производится рубрикация, представляет собой ориентированный граф произвольной сложности, в общем случае – распределенный и многомерный. Никаких количественных ограничений на сложность структуры данных и связей, типы данных и их объем, размерность графа не накладывается. Реструктуризация базы рубрик является рутинной операцией, сравнимой с внесением изменений в данные этой базы. На рис. 1 изображено графическое представление массива рубрик в виде графа. Цвет узлов обозначает их принадлежность одному из серверов базы рубрик, размер иллюстрирует различие структур данных элементов базы, цветом ребер обозначены различные типы связей между рубриками.

Каждый узел графа имеет, в общем случае, переменное количество ребер, в том числе нулевое. Два узла могут иметь более одного общего ребра, расположенных в одном или нескольких измерениях, при этом ребра могут иметь различное направление. Ребра графа представлены логическими, а не физическими, указателями. Операции манипулирования данными могут производиться не только над отдельными элементами, но и над группами элементов, над базой в целом.

Как сама рубрикация, так и реструктуризация графа рубрик - алгоритмически дорогие и сложные операции. Известно, например, что обход дерева имеет экспоненциальную сложность. Трудоемкость обхода графа произвольной структуры, при отсутствии ограничений на количество ребер в узле вообще с трудом поддается количественной оценке. Очевидно лишь, что затраты ресурсов при этом будут чрезвычайно высоки, и практически неприемлемы для графов с количеством узлов порядка нескольких сотен и выше.

Для преодоления этих ограничений нами был разработан собственный алгоритм рубрикации текстовой информации, а также алгоритм реструктуризации графа рубрик, которые реализованы в виде работающих утилит и, как будет показано ниже, имеют **линейную** по числу узлов трудоемкость обхода графа. Это позволяет обрабатывать графы с миллионами и даже десятками миллионов узлов на обычном персональном компьютере. Более того, при этом гарантируется доступ даже к тем узлам и группам узлов, которые не имеют связей с остальными узлами графа (в частности, если эти связи повреждены). Наконец, трудоемкость обхода графа оказывается линейной **в худшем случае** и для некоторых типов задач может оказаться меньше линейной в разы и даже на порядки! Например, при реальной рубрикации текстовой информации скорость обработки превышала линейную на 2-3 порядка и составляла несколько миллиардов узлов в час при использовании компьютеров с тактовой частотой 200-500 МГц.

3 Алгоритмы рубрикации и реструктуризации данных

Базовый алгоритм, применяемый для рубрикации данных, сознательно выбран предельно упрощенным: каждая рубрика характеризуется набором ключевых слов, связанных отношениями булевой логики и круглыми скобками для указания приоритета операций. Никакого синтаксического, орфографического или, тем

более, семантического анализа рубрицируемых данных не производится. Анализ словоформ также не производится, однако многие словоформы все же опознаются алгоритмом рубрикации, поскольку ключевые слова, характеризующие рубрику, обычно заданы в «усеченном» виде, например, с удаленными окончаниями или суффиксами. Кажущаяся «грубость» алгоритма рубрикации оказывается оправданной, поскольку алгоритм становится очень простым и, следовательно, скоростным. Кроме того, можно повысить качество рубрикации простым увеличением числа ключевых слов и их взаимосвязей (критериев) для любой рубрики. Эту гипотезу подтвердило реальное тестирование алгоритма на коллекции из 3 миллионов интернет-ресурсов (сайтов, веб-страниц) при этом количество рубрик составляло более 7 тысяч [2]. Оказалось, что даже если для рубрики задается относительно простая, короткая описывающая «формула», результаты рубрикации оказываются в целом удовлетворительными, хотя процент ошибок все же ощутим. Например, Сан-Франциско попал во Францию, а на слово **катер** мы получили Е**катер**инбург. Однако усложнение описания рубрики резко улучшает качество, и при достаточно объемном и многокритериальном ее описании дает результаты, близкие к ручному рубрицированию.

Сетевая организация графа рубрик оказалась очень удобной для рубрикации сложно структурированной информации, поскольку связи между ее элементами почти никогда нельзя представить в виде иерархической структуры. Действительно, как определить принадлежность рубрике хотя бы одного такого элемента? По какому критерию? По тематике? По географическому положению? По авторам? Популярности? Алфавиту? По любому из заданных для рубрики критериев! Именно сетевая модель данных позволяет безболезненно создавать или модифицировать сколь угодно сложную их комбинацию. При отсутствии этой возможности обеспечить качество рубрикации оказывается весьма затруднительно.

Таким образом, работа алгоритма рубрикации заключается в последовательном просмотре массива данных на предмет наличия в них комбинации ключевых слов, характеризующих каждую рубрику. При обнаружении соответствия, рубрицируемый элемент приписывается к данной рубрике. Ограничений по количеству рубрик, к которым может быть приписан элемент, не предусматривается.

Очевидно, что такое «лобовое» решение предельно неэффективно, поскольку требует многократного просмотра массива

исходных данных. Однако уже сам факт рубрикации данных способен резко ускорить процесс дальнейшей рубрикации. Например, уже упоминавшаяся рубрика **Автоспорт** может быть получена простым пересечением индексов уже отрубрицированных элементов рубрик **Автомобили** и **Спорт**. Или какой-то из разделов рубрики **Авиация** может быть образован объединением и пересечением индексов рубрик **Самолет** | (**Летательный** & **Аппарат**) и т.д. Отметим, что в качестве исходных данных для массивов индексов могут использоваться как контекстные (содержащие какое-то ключевое слово), так и смысловые рубрики (содержащие сколь угодно сложную комбинацию контекстных рубрик, синонимов, сокращений). Отметим также, что формирование таких рубрик не только не требует полного просмотра исходного массива, но может выполняться вообще без обращения к самим данным! Наконец, столь же просто выполняется рубрикация для таких элементов, у которых **нет** заданного ключевого слова или их комбинации. Эта схема хорошо сочетается и с ручной обработкой, если эксперт сказал, что это **то самое**, хотя там, возможно, нет ни одного ключевого слова.

Преимущества метода:

- высокая производительность, обусловленная простотой алгоритма рубрикации и линейным алгоритмом обхода графа рубрик;
- высокая точность, обусловленная возможностью задания сколь угодно сложной комбинации критериев, характеризующих принадлежность элемента рубрике;
- способность алгоритма к смысловой рубрикации, обусловленная возможностью задания любой комбинации ключевых слов, синонимов, словосочетаний и групп ранее отрубрицированных данных;
- возможность рубрикации при отсутствии полного представления об общей структуре коллекции рубрицируемых документов;
- простота реструктуризации графа данных и повторной их рубрикации по измененному графу;
- не требуется разработка сложных утилит анализа информации, смысловых тезаурусов и т.п.;
- не требуется предварительного обучения и, следовательно, дорогостоящей подготовки обучающих последовательностей экспертами в проблемных областях;
- не требуется знаний о языке, на котором написан текст, рубрикация возможна при низком качестве исходной

информации (текстовая, нетекстовая, в различных кодировках, мультиязычная).

Информационные потребности пользователей разные, и они изменяются со временем. Соответственно, структура рубрикатора должна быть гибкой и, по возможности, настраиваемой на конкретного пользователя. Данное требование предполагает наличие механизма эффективной модификации, в том числе реструктуризации, самого рубрикатора. Разработанный нами алгоритм реструктуризации имеет линейную по числу узлов сложность обхода графа. Основная идея состоит в том, что граф рассматривается как таблица неоднородных кортежей (если пользоваться терминами реляционной модели данных). Такой подход позволяет обходить граф последовательно, элемент за элементом, «не обращая внимания» на связи между ними. В действительности трудоемкость обхода оказывается обычно заметно ниже линейной, поскольку имеется возможность просматривать не все, а лишь необходимые элементы (в этом смысле граф рассматривается как отношение со встроенными индексами).

Очень тесно к проблеме рубрикации примыкает проблема наглядного представления данных, в первую очередь, их каталогизация. Представление данных в виде системы каталогов резко повышает наглядность информации, что особенно важно для пользователей с относительно слабой подготовкой. Каталог одновременно является средством эффективного поиска информации без составления специальных запросов к СУБД или средством уточнения таких запросов. Известно, что поисковая система иногда выдает в ответ на запрос пользователя несколько десятков тысяч, а то и миллионов документов. Каталогизация результатов поиска способна заметно облегчить работу с ними. Наконец, наглядность представления информации повышает вероятность визуального выявления ошибок в данных.

Поиск информации при сетевой организации данных рубрикатора организован как симбиоз поисковой системы и каталога, поскольку появляется возможность представления найденных ресурсов пользователю вместе с их связями с другими рубриками (элементами), возможность вести поиск не только по ресурсам, но и по разделам. Возможно сохранение результатов поиска в виде «кармана», (подмножества графа рубрик, хранящее результаты поиска), последующее уточнение результатов запросов. Сам поиск, так же, как и рубрикация, нередко может осуществляться без непосредственного обращения к самим данным, простым

пересечением и/или объединением массивов индексов элементов соответствующих рубрик. При этом скорость поиска многократно возрастает без потери качества. Просто и органично реализуется также возможность коррекции запросов: объединение или пересечение результатов текущего запроса с «карманом» по сложным запросам, поиск внутри «кармана» (поиск в найденном).

Задача рубрикации сильно осложняется наличием ошибок в данных. Опечатки, разное количество пробелов, изменение порядка слов в предложении, использование сокращений и т.п. приводит к появлению фактических дублей и в ряде случаев к лавинообразному росту паразитных записей. Однако та же рубрикация помогает выявлять эти ошибки, иногда довольно сложные. Данные объединяются в группы, в которых «скапливаются» данные, содержащие ошибки. Простейшим примером рубрикации такого рода является выделение уникальных значений полей столбца реляционной таблицы в отдельный столбец для группового исправления ошибок в данных.

Идея состоит в том, что в данных часто встречаются однотипные ошибки. Иногда их оказывается настолько мало, что ошибки можно исправить даже ручным способом. Предположим, в некоторой коллекции данных, содержащей значительное количество имен собственных, 100 раз встречается значение **ААлександр** (двойное нажатие клавиши при набивке оператором), 200 раз **Адександр** (нажатие соседней клавиши), и 300 раз **Алекандр** (клавиша не пропечаталась). После выделения уникальных значений, каждое из этих слов будет встречаться в справочном массиве лишь однажды, и их нетрудно исправить вручную (особенно, если модуль нечеткого поиска укажет на эти слова). После обратного преобразования все 600 ошибок будут исправлены.

В рамках дорожки по классификации РОМИП-2004, помимо основного задания, нами было выполнена организация классификатора компании КОДЕКС в виде графа с сетевой организацией данных (1740 рубрик) [3], а также выявлен ряд ошибок в наиболее популярных словах документов, представленных для рубрикации [4].

4. Заключение

Изложенные выше подходы были применены для разработки ряда продуктов и технологий:

- 2001 г. Технология автоматической рубрикации данных (опробована на БД из 3 миллионов узлов, сформировано более 7000 рубрик) [2];
- 2002 г. Технология динамической коррекции запроса клиента (опробована на БД из 800 тысяч документов при разработке системы электронного документооборота для префектуры СЗАО г. Москвы).
- 2003 г. Пакет программ верификации данных, опробован на реальных базах данных системы электронного документооборота для префектуры СЗАО г. Москвы (около миллиона записей). Выявлено и исправлено более 20 тысяч ошибок в данных.

Предложенные нами принципы рубрикации данных доказали свою жизнеспособность и, пройдя путь через макетное моделирование, послужили основой для разработки ряда программных продуктов.

Литература

[1] В.И. Шабанов А.М. Андреев «Метод классификации текстовых документов, основанный на полнотекстовом поиске»

[2] Распределенный каталог Интернет-ресурсов «Синдбад» http://www.chat.ru/~aleart/0.HTM

[3] Классификатор компании КОДЕКС в виде графа с сетевой организацией данных: http://www.2bit.ru/KODEKC/

[4] Рубрикатор ошибок в наиболее популярных словах документов компании КОДЕКС: http://www.2bit.ru/KODEKC/errors.zip