Правила перезаписи поисковых запросов протокола Z39.50

© Хохлов Александр

Московский государственный университет
alex@lib.msu.ru

Аннотация

При реализации систем распределенного поиска одним из этапов является преобразование поискового запроса из исходной формы в тот вид, который приемлем для каждой конкретной поисковой системы, участвующей в поиске. При этом возможности и форматы запросов в системах различаются, поэтому преобразование приводит к искажениям результатов поиска.

В данной статье рассматриваются правила перезаписи поисковых запросов для протокола Z39.50 на основе диагностических сообщений, приводящие поисковый запрос к поддерживаемому виду для конкретного сервера при минимальном искажении результата поиска в терминах минимального расширения множества результатов поиска.

1 Введение

В настоящее время создается большое количество электронных библиотек. Для обеспечения поиска по всем электронным библиотекам используется либо механизм сбора метаданных для их последующей индексации и поиска (например, с использованием протокола OAI-PMH [1]), либо создается система распределенного поиска или мета-поиска (например, с использованием протоколов Z39.50 [2] или SRW/SRU [3]).

При распределенном поиске по нескольким электронным библиотекам часто возникает ситуация, когда разные библиотеки используют различные схемы данных и различные возможности поиска. Для решения данных проблем либо производится стандартизация поисковых возможностей участвующих в поиске систем [4], либо строится механизм перезаписи запросов в ручном или автоматическом режиме [5].

Труды 7ой Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» - RCDL'2005, Ярославль, Россия, 2005.

В данной статье разбирается задача распределенного поиска с использованием протокола Z39.50 и формальный метод перезаписи type-1 запросов этого протокола, основанный на формальных методах перезаписи булевых запросов [5], позволяющих получать результат поиска в системах, не поддерживающих семантику исходного поискового запроса. При этом множество результатов получаемого запроса должно полностью содержать результат исходного запроса и быть его минимальным расширением.

2 Формальное описание языка запросов протокола Z39.50

2.1 Язык запросов type-1

Язык запросов type-1 в протоколе Z39.50 является булевым языком запросов. В качестве листьев в дереве запроса используется либо терм и атрибутное множество, либо предыдущее результирующее множество или его ограничение.

В виду очень редкого использования в существующих реализациях протокола Z39.50 оператора близости (Proximity operator), а также результирующих множеств (ResultSetId, ResultSet Restriction) в качестве листьев дерева булева запроса type-1, в данной статье они исключаются из рассмотрения. Необходимо отметить, что, даже при наличии реализации результирующих множеств на стороне сервера, их использование на стороне клиента сомнительно по причине зависимости выполнения данного поискового запроса от сеанса Z39.50, в котором было получено используемое результирующее множество. Это жестко связывает сеанс Z39.50 с конкретным пользователем, что, в свою очередь, не позволяет использовать один сеанс Z39.50 между разными пользователями и организовать пул соединений для повышения производительности в многопользовательской среде.

Термом в поисковом запросе является определенное поисковое значение, введенное пользователем. В протоколе оно может представлять собой значение одного из нескольких заранее определенных типов данных, однако здесь рассматривается только тип данных General,

Атр.	Значение	Семантика
1	*Номер точки доступа (поле):* 4 1004 31 …	*Термин должен быть частью указанного поля документа:* Заглавие Автор Дата …
2	*Отношение*: 1 2 3 4 5 6 104	*Термин должен быть в указанном отношении со значением в документе:* < <= = >= > != «между» (для поиска по интервалу)
3	*Позиция:* 1,2 3	*Термин должен быть в следующем месте поля документа, указанной атрибутом 1:* Первый в поле Любое местоположение
4	*Структура*: 1 2 4 6	*Термин представляет собой следующую структуру:* Фраза Слово Год Слова, разделенные пробелами
5	*Усечение*: 1 2 3 100 101	*Термин представляет собой следующее усечение значения в документе:* Правое усечение Левое усечение Левое и правое усечение Нет усечения Использовать символ # в термине
6	*Полнота*: 1 2,3	*Термин должен совпадать с:* Неполным значением поля документа Всем значением поля документа

Таблица 1. Семантика type-1 запросов в атрибутном множестве bib-1

передающий значение в виде текстовой строки. Другие типы данных, такие как Numeric или DateTime, применимы только для версий протокола Z39.50, начиная с третьей, и, в большинстве случаев все равно приводятся сервером к текстовому виду при выполнении поискового запроса.

Атрибутное множество представляет собой множество пар вида (attribute, value), которые задают значения атрибутам терма из заранее определенной схемы. Здесь рассматривается наиболее часто используемая атрибутная схема bib-1 [6], поэтому атрибут и значение являются числами.

В результате BNF грамматика разбираемых в данной статье type-1 запросов следующая:

```
  RPN-Query ::= Argument | (Argument
Operator Argument)
  Argument ::= Operand | RPN-Query
  Operand ::= (AttributeList Term)
  Operator ::= AND | OR | AND-NOT
  AttributeList ::= | AttributeList
Attribute=Value
  Attribute ::= число
  Value ::= число
  Term ::= строка
```

Пример 1.

Пусть пользователь ввел для поиска в поле «автор» значение «Пушкин», а в поле «заглавие» значение «Евгений Онегин». Пусть между полями подразумевается булев оператор «И», а поиск терма «Евгений Онегин» подразумевается как фразы в пределах поля «заглавие».

Тогда type-1 запрос для данного поискового запроса будет следующим:

```
(
  (
    (1=1004 2=3 3=3 4=2 5=100 6=1
     "Пушкин")
  ) AND (
    (1=4 2=3 3=3 4=1 5=100 6=1
     "Евгений Онегин")
  )
)
```

2.2 Вычисление запросов type-1

Вычисление описанного дерева булевых запросов выполняется снизу вверх. При построении множества результатов, удовлетворяющих листу булева запроса (AttributeList Term) при использовании атрибутного множества bib-1, используется семантика [7] (см. Таблицу 1). Заметим, что не все возможные значения атрибутов и их комбинаций рассматриваются при преобразовании.

Например, следующая запись*: (1=4 2=3 3=1 4=1 5=1 6=1 алгоритмы)*, означает: поиск терма «алгоритмы», в поле «заглавие», на равенство, с правым усечением, в начале поля, с неполным совпадением.

Список стандартных комбинаций атрибутов, которые являются обязательными для реализации в рамках протокола Z39.50, определяются различными профилями протокола. Для библиотечных приложений наиболее широкое распространение получили профили NISO Z39.89 [4] и BATH [8], а также производные от них.

3. Перезапись запросов при отсутствии необходимой функциональности

Наличие обязательных для исполнения профилей, упомянутых ранее, не означает, что все реализации протокола Z39.50 им удовлетворяют. Существует достаточно большое количество реализаций, включая типовые решения в рамках комплексных библиотечных систем, которые нарушают те или иные требования.

Кроме того, различные источники данных могут поддерживать различные точки доступа и их разную детализацию (т.е. наличие кроме поля «автора» двух полей «персональный автор» и «коллективный автор»).

Поэтому для корректной работы требуется приводить запрос к той форме, которая поддерживается конкретной реализацией протокола Z39.50 и учесть изменения результата поиска, возникающие в результате изменения запроса.

3.1 Мета-информация о поддерживаемых возможностях поиска

При существующих различиях в реализациях протокола Z39.50 для преобразования запросов необходима информация о поддерживаемых точках доступа и возможностях поисковых запросов. Эту информацию можно извлечь вручную (проведя тестирование исследуемого сервера) или использовать службу Explain протокола Z39.50 [2].

Однако в рамках протокола Z39.50 предусмотрено возвращение клиенту диагностических кодов ошибок в случаях, когда определенные значения атрибутов не поддерживаются данным сервером. Этот механизм позволяет получать необходимую мета-информацию о поисковых возможностях сервера в процессе работы без его предварительного анализа.

3.2 Возможные ограничения функциональности поиска

При работе по протоколу Z39.50 сервер среди прочих может вернуть следующие сообщения из диагностического множества diag-1 (в круглых скобках указывается код сообщения):

- «указанное значение атрибута не поддерживается» (один из 114, 117, 118, 119, 120, 122)
- «указанный атрибут не поддерживается вовсе» (113)
- «не поддерживается указанная комбинация атрибутов» (123) или «поисковый запрос не поддерживается» (3)

Заметим, что диагностическая ошибка описывает только тип возникшей проблемы, но не указывает на то, в какой части поискового запроса она возникла (если поисковый запрос состоит из множества листьев, соединенных булевыми операторами).

Из-за отсутствия информации о фактическом местоположении проблемного значения атрибута при реализации механизма перезаписи запросов применяются преобразования, влияющие на весь запрос (меняющие все возможные проблемные места).

Для более точной локализации проблемы в поисковом запросе его выполнение можно разбить на несколько этапов, где первый запрос будет содержать первый лист type-1 запроса, а каждый следующий будет добавлять в качестве уточнения новый лист к полученному на предыдущем шаге результирующему множеству с соответствующей булевой операцией.

Работа с результирующими множествами в данном контексте не повлияет отрицательно на работу в многопользовательской среде (как было отмечено ранее для общего случая), но последовательное выполнение запроса существенно увеличит количество поисковых запросов и ответов между клиентом и сервером, что существенно повлияет на скорость выполнения всего запроса в целом. Кроме того, как уже отмечалось, в виду редкости реализаций результирующих множеств их применение на стороне клиента сильно ограничено.

3.3 Правила перезаписи запросов

Для того, чтобы привести поисковый запрос Q в соответствие поддерживаемым поисковым возможностям сервера, будем преобразовывать поисковый запрос Q в Q' такими преобразованиями, чтобы множество результатов $R(Q)$ содержалось в $R(Q')$. Для этого преобразуем поисковый запрос Q в ДНФ и каждый лист L получившегося булева дерева будем заменять на поддерживаемый поисковым сервером лист L' такой, чтобы $R(L)$ содержалось в $R(L')$. Получившийся запрос Q' будет

поддерживаться поисковым сервером и, как отмечается в [5], будет минимально расширять результирующее множество, если каждое из преобразований листьев *L* будет минимальным образом расширять результат *R(L)*.

Так как каждый лист в type-1 поисковом запросе представляет собой пару «набор значений атрибутов» - «значение терма», то минимальной единицей изменения type-1 запроса будет именно эта пара.

Пример 2.

Рассмотрим тот же поисковый запрос, который был приведен в примере 1 («автор» = «Пушкин» И «заглавие» = «Евгений Онегин»).

Предположим, что поиск фразы в поле «заглавие» не поддерживается поисковым сервером, и поэтому запрос в исходном виде не может быть выполнен. В случае, если по полю «заглавие» можно проводить поиск слов, то поисковый запрос можно преобразовать к виду: «автор» = «Пушкин» И «заглавие» = «Евгений» И «заглавие» = «Онегин».

Измененный поисковый запрос имеет потенциально более широкое множество результатов, так как не учитывает порядок слов «Евгений» и «Онегин» в поле «заглавие». Однако модифицированный поисковый запрос будет исполнен, а лишние результаты поиска можно, при желании, отфильтровать на стадии извлечения и показа результатов конечному пользователю.

Рассмотрим далее с учетом перечисленных возможных диагностических сообщений различные ситуации, возникающие при работе с различными реализациями протокола Z39.50.

Так как диагностические сообщения относятся к атрибутам, то для каждого атрибута и его значения необходимо построить два преобразования: одно должно минимально увеличивать результат поиска, а другое – быть его минимальным сужением. Второе преобразование необходимо, когда лист в ДНФ(*Q)* встречается «под» операцией AND-NOT.

В процессе преобразования может возникнуть необходимость задания «вырожденных» листов, которые отвечают результирующим множествам «пусто» и «все документы». Условимся такие листы обозначать NONE и ALL соответственно.

3.3.1 Неподдерживаемое значение атрибута точки доступа (114)

В этом случае поисковый сервер не поддерживает поиск по указанной точке доступа. Здесь для приведения поискового запроса к поддерживаемому виду необходимо заменить точку доступа на другую, которая поддерживается сервером.

Заметим, что точки доступа в атрибутной схеме bib-1 [6], а также и в других атрибутных схемах, разработанных и используемых в рамках протокола Z39.50, можно представить в виде иерархической системы по точности задания семантики поля. В такой иерархии в корне стоят равнозначные значения 1016 (любое) и 1035 (везде), а все остальные точки доступа являются уточнением непосредственно верхних по иерархии. Например, иерархия полей, связанных с автором, может быть следующая: 1016 (любое) → 1003 (автор) → 1004 (персональное имя). Иерархия для наиболее часто используемых атрибутов схемы bib-1 приведена в таблице 2.

Таким образом, для преобразования запроса к поддерживаемой форме с минимальными изменениями результата поиска необходимо сделать следующие замены:

- Если есть равнозначная точка доступа (синоним), то попробовать заменить точку доступа на синоним, иначе применить следующие два правила.
- Для минимально расширяющего преобразования: если это не 1016 или 1035, то заменить точку доступа на непосредственно верхнюю по иерархии уточнения поля, иначе заменить лист на ALL.
- Для минимально сужающего преобразования: заменить точку доступа на пересечение поиска термина по всем непосредственно дочерним точкам доступа в иерархии уточнения поля, если таковые имеются, иначе заменить лист на NONE.

3.3.2 Неподдерживаемое значение атрибута отношения (117)

В этом случае поисковый сервер не поддерживает поиск по указанному отношению.

- Если значение отношения равно 104 (между), то его можно попробовать преобразовать к эквивалентному подзапросу с «>= И <=».
- Если же значение отношения не равно 104, то, к сожалению, в данной ситуации минимальным расширением и сужением может служить только лист ALL и NONE соответственно. На практике это означает, что если сервер не поддерживает поиск по какому-либо оператору отношения, то этот лист поискового запроса должен быть полностью исключен из запроса.

3.3.3 Неподдерживаемое значение атрибута позиции (119)

Если не поддерживаются эквивалентные между собой значения 1 или 2 (поиск в начале поля), то необходимо произвести следующие замены: для минимально расширяющего преобразования – поставить значение 3 (поиск в любом месте поля), для минимально сужающего преобразования – лист NONE.

Если не поддерживается значение 3, то минимально расширяющим преобразованием является ALL, а минимально сужающим – замена значения на 1.

Корень иерархии	Основная категория	Дополнительные категории
1016 (любое) =1035 (везде)	1003 (автор)	2=1005 (наименование организации)
		3=1006 (коллективный автор)
		1=1004 (персональное имя)
	4 (заглавие)	5 (серия)
		6 (общее заглавие)
		33 (ключ заглавия)
		34 (собирательное заглавие)
		35 (параллельное заглавие)
		36 (заглавие с обложки)
		37 (подзаголовочные данные)
		38 (основное заглавие)
		39 (колонтитул)
		40 (заглавие на корешке)
		41 (альтернативное заглавие)
		42 (предыдущее заглавие)
		43 (сокращенное заглавие)
		44 (расширенное заглавие)
	21 (тема)	22 (тема RAMEAU)
		23 (тема BDI)
		24 (тема INSPEC)
		25 (тема MESH)
		26 (тема PA)
		27 (тема LC)
		28 (тема RVM)
		29 (тема по локальной схеме предметизации)
		45 (тема PRECIS)
		46 (тема RSWK)
		47 (подраздел темы)
		1009 (тема – имя персоналии)
	30 (дата)	31 (дата публикации)
		32 (дата приобретения)
		1011 (дата добавления в базу данных)
		1012 (дата последней модификации)
	59 (место публикации)	
	…	

Таблица 2. Иерархия некоторых значений атрибута 1 (поле) атрибутной схемы bib-1

3.3.4 Неподдерживаемое значение атрибута структуры (118)

Если не поддерживается значение 4 (год), то его можно заменить на «почти» эквивалентное значение 2 (слово). Результат поиска будет таким же, так как поиск по году все равно происходит в текстовом виде и значение 4 (год) не используется реализациями протокола Z39.50 для каких-либо семантических изменений выполнения поисковых запросов.

Если не поддерживается значение 2 (слово), то минимальным расширением будет ALL, а минимальным сужением будет замена на значение 1 (фраза).

Если не поддерживается значение 1 (фраза), то минимальным расширением будет разбиение фразы на слова и замена листа на конъюнкцию листов со значениями атрибута структуры 2 (слово), а минимальным сужением будет NONE.

Если не поддерживается значение 6 (список слов), то его можно заменить на эквивалентный подзапрос, составленный разбиением введенного термина на отдельные слова по символу «пробел» и объединению отдельных терминов оператором «И».

3.3.5 Неподдерживаемое значение атрибута усечения (120)

Если не поддерживается значение 1 / 2 (левое / правое усечение), то минимальным расширением будет значение 3 (левое и правое усечение), а минимальным сужением – 100 (без усечения).

Если не поддерживается значение 3 (левое и правое усечение), то минимальным расширением будет значение ALL, а минимальным сужением – конъюнкция двух листов с тем же термином, но значениями атрибута усечения 1 и 2.

Если не поддерживается значение 100 (без усечения), то минимальным расширением будет конъюнкция двух листов с тем же термином, но значениями атрибута усечения 1 и 2, а минимальным сужением – NONE.

Если не поддерживается значение 101 (использовать символ # в значении термина), то возможно заменить лист на подзапрос, который будет составлен исходя из положения символа # в термине: необходимо объединить оператором И поиск по началу и окончанию поискового термина (с соответствующими значениями атрибута усечения) до ближайшего символа #. Подзапрос с пустой частью термина (когда символ # стоит в начале или в конце) необходимо сразу отбросить. Если символ # встречается и в конце, и в начале, но его нет в середине термина, то заменить атрибут на значение 3 (левое и правое усечение), соответственно убрав из термина все символы #.

3.3.6 Неподдерживаемое значение атрибута полноты (122)

Если не поддерживается значение атрибута 1 (неполное значение поля), то минимальным расширением будет ALL, а минимальным сужением – значение 2 или 3.

Если не поддерживается значение атрибута 2 или 3 (полное значение подполя или поля), то минимальным расширением значение 1, а минимальным сужением – лист NONE.

3.3.7 Какой-либо атрибут полностью не поддерживается сервером (113)

В этом случае возможно два варианта: либо исключение атрибута (но не листа) из поискового запроса, что приведет к непредсказуемому изменению результата поиска (так как нельзя узнать, какое все-таки значение данного атрибута подразумевается сервером), либо его замена на ALL или NONE (т.е. полное исключение данного листа из запроса).

Заметим, что исключение атрибута с непредсказуемыми последствиями возможно и во всех предыдущих разобранных случаях (3.3.1 – 3.3.6), когда должна происходить замена листа на ALL или NONE.

Решение об исключении листа или исключении только атрибута из листа необходимо принимать на основе предположения о том, что при исключении атрибута сервер скорее будет подразумевать необходимое в данном случае значение атрибута.

Например, если сервер не поддерживает атрибут усечения, а в листе он равен 100 (без усечения), то логично будет исключить только атрибут усечения. При других значениях атрибута усечения более оправдано будет исключение всего листа из запроса и его замены на лист ALL или NONE в зависимости от контекста вхождения листа в запрос.

3.3.8 Не поддерживается указанная комбинация атрибутов (123) или поисковый запрос не поддерживается (3)

Данный вид диагностического сообщения не предоставляет информации о том, какой именно лист стал причиной данного вида проблемы, поэтому данный вид диагностической ошибки необходимо считать фатальным, либо принимать решения на основе информации о наиболее часто неподдерживаемых значениях атрибутов, использованных в запросе.

Так, при использовании атрибута отношения 104 (между), многие сервера отвечают именно этим диагностическим сообщением. Замена атрибута отношения 104 (между) на подзапрос «>= И <=», часто дает положительный результат.

Точные сведения о том, как преобразуется результат поиска при изменении запроса, зависит от тех действий, которые предпринимаются в ответ на данное диагностическое сообщение.

4. Оценка точности и полноты модифицированного запроса

Хотя описанные методы производят лучшее приближение исходного результата поиска в терминах минимального расширения результирующего множества при операциях модификации запроса, они не позволяют провести оценку точности и полноты модифицированного поиска по сравнению с исходным запросом.

Более того, в ряде случаев поисковый запрос вырождается в пустое множество или во множество со всеми документами, т.е. в худшем случае результат кардинально изменяется. В лучшем случае он преобразуется в эквивалентный, и тогда его точность и полнота по сравнению с исходным запросом не изменяются.

Однако в большинстве случаев вырождение запроса происходит в тех местах, где невозможность выполнения исходных условий поискового запроса приводит к полной невозможности его выполнения. Это означает, что не существует метода его формулирования в приемлемом для исполнения виде, который бы коррелировал с исходным поисковым запросом, и потеря полноты и точности оправдана.

Правила перезаписи запросов, которые здесь рассмотрены, предназначены для случая преобразования запроса без потери релевантных ответов, но с появлением дополнительного шума. Возможно поставить задачу иначе: модифицировать поисковый запрос так, чтобы были возвращены только релевантные ответы, возможно с их частичной потерей.

В этой постановке задача решается тем же методом, только при преобразовании каждого листа необходимо использовать процедуру, производящую минимальное сужение множества результатов поиска. Тем самым данная задача

является зеркальной по отношению к той, которая рассматривалась в данной работе, и в ней применимы все описанные преобразования.

5 Заключение

В данной статье приводятся правила перезаписи поисковых запросов протокола Z39.50, которые приводят запрос в вид, поддерживаемый конкретным поисковым сервером, с которым открыт сеанс поиска.

К сожалению, правила в ряде случаев вырождаются в пустые запросы, либо преобразуются в запросы, судить об изменении результата поиска которых по сравнению с исходным запросом невозможно. Однако эти случаи отражают ситуации, где различия в семантике исходного поискового запроса и в возможностях рассматриваемого сервера Z39.50 настолько серьезны, что выполнение поискового запроса, близкого к исходному, не представляется возможным.

Тем не менее, указанный механизм перезаписи запросов позволяет в большинстве случаев в автоматическом режиме настраиваться системе распределенного поиска на различия в поисковых возможностях серверов Z39.50 и предоставлять пользователю информацию об изменениях семантики поисковых запросов.

В дополнение к описанной схеме перезаписи поисковых запросов протокола Z39.50 можно добавить механизм пост-фильтрации результатов поиска для приведения семантики результата поиска к семантике исходного поискового запроса на стадии извлечения результатов поиска и их представления конечному пользователю [5].

Литература

1. The Open Archives Initiative. *Protocol for Metadata Harvesting (OAI-PMH)*. Protocol Version 2.0, 2002 http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm
2. National Information Standards Organization. *Information Retrieval: Application Service Definition and Protocol Specification* (ANSI/NISO Z39.50). NISO Press, Bethesda, Md., 2003. http://www.niso.org/standards/resources/Z39-50-2003.pdf
3. *Search/Retrieve Web Service (SRW)*. Version 1.1 http://www.loc.gov/z3950/agency/zing/srw/
4. National Information Standards Organization. *The U.S. National Z39.50 Profile for Library Applications (ANSI/NISO Z39.89)*. NISO Press, Bethesda, Md., 2003. http://www.niso.org/standards/resources/Z39-89final.pdf
5. K. C.-C. Chang. *Query and Data Mapping Across Heterogeneous Information Sources.* PhD thesis, Stanford Univ., January 2001 http://eagle.cs.uiuc.edu/pubs/2001/thesis.ps.gz
6. Bib-1 Attribute Set, 2003 http://www.loc.gov/z3950/agency/defns/bib1.html
7. Attribute Set Bib-1 (Z39.50-1995): Semantics. 1995 http://www.loc.gov/z3950/agency/bib1.html
8. The Bath Group. *The BATH Profile: An International Z39.50 Specification for Library Applications and Resource Discovery*. Release 2.0, 2003 http://www.nlc-bnc.ca/bath/tp-bath2-e.htm

Rules for search query rewriting in Z39.50 protocol

Khokhlov Alexandre

Query translation from the user form to the form appropriate for each database is one of the stages of query evaluation in distributed search systems. Databases differ in their search capabilities and data formats, and this leads to distortion of result sets.

This article discusses the rules for query rewriting of search requests within the Z39.50 protocol. The rules are applied according to diagnostic messages sent back by server and are minimal in terms of result set widening from the true one while making the query acceptable by server.