Преобразование запросов для предметно-независимого фактографического поиска в Интернет

© Илья Бояндин, Игорь Некрестьянов

Санкт-Петербургский Государственный Университет
http://ir.apmath.spbu.ru

Аннотация

В работе рассматривается задача преобразования вопросов, задаваемых пользователями предметно-независимой системы фактографического поиска в Веб на естественном языке, в запросы поисковой системы общего назначения.

Анализируется работа статистического алгоритма преобразования запросов, основанного на модели QASM [17]. Оценка качества поиска с использованием предлагаемого алгоритма производится на реальных русскоязычных фактографических запросах из журнала поисковой системы Яндекс.

1 Введение

Задача фактографического поиска — это разновидность задачи текстового поиска с уменьшенной гранулярностью [2]. В отличие от классической задачи поиска при фактографическом поиске необходимо обнаружить не документы на тему запроса, а точные и лаконичные ответы на конкретные вопросы, сформулированные на естественном языке. Например, на вопрос: "Кто был первым космонавтом?" идеальная система фактографического поиска должна выдавать единственный ответ: "Юрий Гагарин".

В отличие от вопросно-ответных систем с активным усвоением знаний от систем предметно-независимого фактографического поиска не требуется способность производить логический вывод - система должна лишь выделить из набора данных короткий фрагмент текста, который является ответом на вопрос. Поэтому результат работы такой системы сильно зависит от набора текстовых данных, в которых производится поиск. Например, если поиск ответа на тот же вопрос о первом космонавте производится в коллекции текстов об американской космонавтике, то правильным ответом вполне может оказаться "Алан Шепард".

На данный момент в Интернет отсутствуют промышленные системы, способные автоматически обрабатывать фактографические запросы с приемлемым качеством. Но потребность в обработке таких запросов несомненно существует. Согласно результатам анализа журнала поисковой системы Excite[1] около 8% пользовательских запросов являются корректными вопросами английского языка, из них около 44% — фактографическими [4]. Пользователи русскоязычной поисковой системы Яндекс тоже нередко формулируют свои запросы в виде корректных вопросов [3].

Актуальность задачи фактографического поиска стимулирует активные исследования в этой области. В частности, в рамках конференции TREC уже несколько лет существует отдельная дорожка, посвященная экспериментальной оценке систем фактографического поиска [19].

Огромный объем постоянно обновляемой и пополняемой текстовой информации в Интернет дает системам фактографического поиска потенциальную возможность находить ответы на гораздо более разнообразные фактографические вопросы, чем это возможно в рамках закрытой коллекции документов.

Однако данные в Интернет из-за отсутствия централизованного контроля над публикацией обладают рядом особенностей, осложняющих решение задачи поиска: неструктурированностью, разнородностью, противоречивостью. С другой стороны, в Интернет также наблюдается избыточность, повторяемость данных: ответ на один и тот же вопрос, по-разному сформулированный, может содержаться в десятках документов. Как было отмечено рядом исследователей, эта избыточность может быть использована для повышения качества фактографического поиска [11, 9], что косвенно подтверждают результаты экспериментов по выявлению зависимости качества фактографического поиска от размера коллекции данных [7, 8].

Обработка запроса в системе фактографического поиска обычно производится в несколько этапов [15]:

1. **Предварительный**
   На этом этапе обычно выполняется классификация вопроса, на основе результатов которой, формулируется запрос и определяются возможные формы ответов.

[1] за 20 декабря 1999 года

Proceedings of the 5th Russian Conference on Digital Libraries RCDL2003, St.-Petersburg, Russia, 2003

2. **Преобразование вопроса**
   Вопрос преобразуется в один или несколько запросов поисковой системы, так чтобы найденные документы как можно точнее представляли документы коллекции, в которых содержатся возможные ответы.

3. **Текстовый поиск**
   Поисковая система, используя традиционные методы информационного поиска, находит документы коллекции, соответствующие сформулированным запросам.

4. **Выделение ответов**
   Из найденных поисковой системой документов выбираются небольшие фрагменты текста, содержащие наиболее вероятные ответы. Системы пытаются различными способами убедиться в правильности каждого из возможных ответов-кандидатов и отбросить "неубедительные".

Существующие в Интернет поисковые системы общего назначения могут быть эффективно использованы системой фактографического поиска на этапе текстового поиска для обнаружения документов с возможными ответами [16]. Например, запрос "`Набоков /1 !родился`", посланный Яндекс, дает пять правильных ответов (в 1899 году) из первых пяти, тогда как среди первых пяти документов, полученных по запросу "`Когда родился Набоков?`", только один содержит правильный ответ[2]. Качество преобразования вопроса в запрос поисковой системы в значительной мере определяет общее качество поиска. Действительно, если ни один документ, содержащий искомый ответ, не будет найден при помощи построенных запросов, то последующие этапы никак не смогут исправить ситуацию; если же несколько найденных документов будут содержать правильный ответ, это повысит вероятность его выбора системой на этапе выделения ответов[3].

Целью этой работы было исследование возможности эффективного преобразования вопросов на основе статистических методов. На сегодняшний день наилучшие результаты фактографического поиска обеспечивают системы, активно использующие методы работы с текстами на естественном языке, но как показал пример системы Tritus[4], системы использующие статистические методы и очень простые знания о языке, могут достигать значительных результатов.

Использование статистических подходов зачастую позволяет значительно снизить вычислительную трудоемкость обработки запроса и, как следствие, повысить масштабируемость системы. Другое важное преимущество статистического подхода — уклонение от необходимости использования высококачественных лингвистических ресурсов, доступность которых, например, для русского языка ограничена.

В качестве отправной точки наших исследований был использован алгоритм QASM [17], который обучается преобразованиям вопросов на наборе вопросов и ответов. Начав с модели QASM, мы попытались найти ответы на ряд вопросов: какова максимальная достижимая эффективность преобразований в этой модели? насколько отличается эффективность операторов и свойств? от каких факторов зависит оптимальный выбор?

Дальнейшее изложение имеет следующую структуру: в разделе 2 представлен обзор близких работ, в разделе 3 описан базовый алгоритм QASM, а в разделе 4 — используемые нами модификации алгоритма. В разделе 5 изложены наиболее важные характеристики прототипа системы фактографического поиска, анализ результатов экспериментов с которым представлен в разделе 6.

2 Близкие работы

Преобразование запросов активно используется во многих задачах, связанных с поиском информации. По типу цели преобразования их можно разделить на два основных класса:

- **"Перевод" запроса**
  К этой группе относятся преобразования, которые предназначены для выражения того же запроса в другом виде с максимальным сохранением свойств исходного запроса. Например, к этому классу относятся преобразования запросов посредником в метапоисковых системах (где каждый из поисковых серверов может иметь свой язык запросов со специфичным синтаксисом и семантикой) [14] или в системах многоязычного поиска (где исходный запрос может автоматически переформулироваться на другом языке) [20].

- **"Уточнение" запроса**
  Преобразования этого вида изначально ориентированы на изменение свойств запроса, то есть его семантики. Целью этого изменения обычно является получение новой редакции запроса, которая лучше описывает информационную потребность, стоящую за исходным запросом. Такие преобразования, например, часто используются вместе с механизмом обратной связи (relevance feedback) для итеративного уточнения поисковой системой потребностей пользователя [5].

В контексте предметно-независимого фактографического поиска особый интерес представляют преобразования второго типа, поскольку поисковые системы общего назначения не предназначены для поиска ответов на вопросы естественного языка, и для того чтобы правильно сформулировать на языке запросов поисковой системы

[2] Описанный эксперимент с Яндекс проводился 15 марта 2003

[3] Во многих системах (например, в системе Mulder [15]) оценки для возможных ответов на этапе выделения ответов вычисляются таким образом, что вероятность выделения ответа, чаще присутствующего в найденных документах, повышается.

Вопрос	Тип вопроса
Как зовут создателя логотерапии?	ЛИЧНОСТЬ
В каком году был построен Мавзолей в Берлине?	ДАТА
Где находится Тадж Махал?	МЕСТО
Каково расстояние от Абу-Даби до Агры?	РАССТОЯНИЕ

Таблица 1: Примеры типов вопросов

реальную информационную потребность пользователя, необходимо изменить семантику исходного вопроса. Например, некоторые слова, содержащиеся в вопросе, совсем не обязательно должны содержаться в правильном ответе на этот вопрос: они могут просто отсутствовать, присутствовать в другой форме или их могут заменять синонимы или обобщающие понятия.

Преобразования вопросов в системах фактографического поиска обычно можно представить в виде последовательности операций, таких как: добавление, удаление или замена слов, фраз или добавление операторов синтаксиса поисковой системы, морфологические преобразования слов и т.п. (см, например, [15]).

Выбор операций преобразования, которые применяются к вопросу, основывается на различных характеристиках вопроса или отдельных слов, участвующих в вопросе. Важнейшим свойством вопроса является его тип. Наборы типов вопросов, используемые разными системами, и методы их определения различаются, но в большинстве систем тип вопроса задается объектом вопроса (см. таблицу 1), который определяется по вопросительным словам (см, например, [4]) или при помощи более сложных синтаксического и семантического разборов вопроса (см., например, [13] или [6]). К характеристикам отдельных слов вопроса, которые используются для определения преобразований, могут относиться: роль слова (например, вопросительное или нет), часть речи, значимость слова (оцениваемая на основе частоты его использования) и др. [17]

Правила преобразования могут быть предопределены в системе заранее с разной степенью обобщенности. Например, в работе [18] формулируются простые запросы, состоящие из наиболее "важных" по статистическим свойствам ключевых слов вопроса. Этот подход обеспечивает невысокую точность, но довольно хорошую полноту поиска[4].

В системе Falcon [13] оптимальная степень ослабления запроса определяется динамически. Получив результаты поиска по начальному запросу, система формулирует ослабленную версию запроса (удаляя некоторые слова), если результатов найдено слишком мало, или формулирует более строгий запрос (добавляя в него термы), если результатов слишком много.

В системе AskMSR [11] правила преобразования задаются вручную. Достаточно строгие правила, созданные вручную, могут обеспечивать высокую точность поиска. Однако поскольку строгие правила чаще всего узкоспециализированы, то есть применимы в весьма ограниченном наборе случаев, то создание исчерпывающего набора таких правил, учитывающего все особенности естественного языка, вряд ли возможно.

В течение нескольких последних лет много внимания уделяется исследованию подходов, автоматически обучающихся преобразованиям запросов (см., например, [12]), в том числе и для фактографического поиска.

Система Tritus [4] обучается правилам преобразования фактографических вопросов на наборе "часто задаваемых вопросов" и ответов на них. При обучении система сначала пытается найти важные фразы, наиболее часто встречающиеся в фрагментах текста, содержащих ответы на вопросы каждого типа, взвешивая фразы с помощью весов, подобных $tfidf$, и строит правила преобразований, используя эти фразы. Затем система оценивает качество всех полученных преобразований, применяя их ко всем вопросам соответствующего типа из тренировочного набора, передавая запросы поисковой системе и оценивая близость первых n найденных документов к известному ей ответу на вопрос, и выбирает и запоминает только наилучшие преобразования.

3 Алгоритм QASM

Вероятностный алгоритм QASM (Question Answering using Statistical Models) [17] обучается преобразованиям, которые представляют собой композиции атомарных преобразований.

Задача алгоритма — построить по входному вопросу последовательность атомарных преобразований, композиция которых в применении к нему дает наилучший запрос[5]. Процедура построения запроса итеративна: на каждом шагу выбирается атомарное преобразование, улучшающее запрос; итерации продолжаются, пока улучшение возможно.

Для того чтобы алгоритм QASM мог строить преобразования вопросов, ему необходимо пройти этап обучения, на котором он выявляет закономерности, связывающие характеристики запроса и удачные преобразования.

[4] Отметим, что такой подход значительно повышает нагрузку на последующие шаги обработки фактографического запроса. Хотя в этом случае система получает больше документов, содержащих правильный ответ, она также получает и значительно большее количество неподходящих документов, которые могут ввести ее в заблуждение.

[5] В этой статье термин *вопрос* используется для обозначения исходного вопроса пользователя на естественном языке, а термин *запрос* чаще всего используется для обозначения преобразованной версии вопроса, которая отправляется поисковой системе.

Вообще говоря, задачу выбора атомарного оператора можно рассматривать как задачу классификации: алгоритм должен решить, к какому классу отнести запрос на данном шагу, и применить к нему атомарный оператор, соответствующий выбранному классу.

3.1 EM-алгоритм

В основе алгоритма обучения QASM лежит известный статистический алгоритм *максимизации ожидания* (Expectation Maximization) — итеративный алгоритм нахождения оценок максимального правдоподобия. Этот алгоритм часто используется в задачах с неполными данными. В нашем случае при обучении известен только набор вопросов и ответов, а сами преобразования, дающие наилучший результат для каждого из вопросов, неизвестны.

EM-алгоритм состоит из следующих шагов:

1. оцениваются неизвестные параметры (используются доступные алгоритму результаты измерений и данные модели),
2. модифицируется модель алгоритма (на основе полученных на шаге 1 оценок),
3. если не достигнут локальный максимум, то переходим к шагу 1.

Известно, что EM-алгоритм с каждым шагом обеспечивает улучшение получаемых оценок и в конце концов сходится [10].

3.2 Обучение

QASM обучается на наборе вопросов и ответов, пытаясь найти для каждого из вопросов тестового набора наилучшую последовательность атомарных преобразований. Формально задачу можно поставить следующим образом.

Пусть $\mathcal{A}_1, \ldots, \mathcal{A}_l$ — фиксированный набор функций (*свойств запросов*), сопоставляющих запросу целое число. Упорядоченный набор численных значений всех этих функций для конкретного запроса q назовем контекстом $\mathcal{C}(q)$ запроса:

$$\mathcal{C}(q) := (\mathcal{A}_1(q), \ldots, \mathcal{A}_l(q))$$

Пусть $\mathcal{O}_1, \ldots, \mathcal{O}_m$ — фиксированный набор *атомарных операторов преобразования* запросов, сопоставляющих запросу q новый запрос $\mathcal{O}_i(q)$, а $F(q)$ — некоторая функция оценки *реального качества* запроса, которая может использовать информацию о найденных по этому запросу документах (см. раздел 5.3).

Отметим, что только оператор $Identity$, сопоставляющий запросу самого себя ($Identity(q) = q$), является фиксированным. Выбор других операторов, свойств и функции $F(q)$ не влияет на общий алгоритм и зависит от его реализации.

Алгоритм обучения должен определить отображение T, ставящее в соответствие любому набору $\mathcal{C}(q)$ значений свойств запроса q номер r атомарного оператора, приводящего к наиболее качественному преобразованию.

Другими словами, алгоритм обучения строит классификатор T, который каждому допустимому контексту сопоставляет класс, соответствующий одному из атомарных операторов, наилучшему для данного контекста.

Отображение T в алгоритме QASM задается матрицей

$$\Theta = \{p(\mathcal{O}_i|\mathcal{C}_j)\}_{i,j}$$

вероятностей $p(\mathcal{O}_i|\mathcal{C}_j)$ применения оператора $\mathcal{O}_i$ к запросу, задающему контекст $\mathcal{C}_j$:

$$T(\mathcal{C}_j) = \underset{i}{argmax}(\Theta_{i,j})$$

где $\mathcal{C}_j$ - допустимый контекст с номером j (число всех допустимых контекстов конечно, поэтому их можно занумеровать). Причем,

$$\forall j : \sum_{i=0}^{m} p(\mathcal{O}_i|\mathcal{C}_j) = 1$$

Матрица Θ инициализируется равномерным распределением по всем операторам. Далее, алгоритм обучения последовательно выполняется для каждого из вопросов тренировочного набора на одной и той же матрице Θ:

1. Применить каждый из операторов к вопросу, оценивая качество получаемых запросов (правильный ответ на исходный вопрос известен). Если наибольшее качество обеспечивает оператор $Identity$, то обработка текущего запроса завершена. Иначе — на основе распределения вероятностей, заданному в Θ для контекста запроса, выбрать оператор, который будет применен к запросу.

2. Применить к запросу выбранный на первом шаге оператор. Модифицировать Θ, используя информацию о качестве запросов, полученную на первом шагу:

   - операторы упорядочиваются по убыванию качества получаемых запросов (для данного запроса q),
   - вероятности $p(\mathcal{O}_i|\mathcal{C}(q))$ в строке Θ, соответствующей $\mathcal{C}(q)$, домножаются на $oprank_i^{-1}$, где $oprank_i$ — ранг i-го оператора (присвоенный ему в результате упорядочивания),
   - строки матрицы нормализуются, так чтобы сумма значений в строке равнялась 1.

3. Если изменение матрицы Θ на этой итерации не превышает заданного порога ($\delta(\Theta) < \varepsilon$), то обработка текущего запроса завершена, а построенная матрица Θ и есть результат обучения алгоритма. Иначе цикл повторяется с первого шага.

3.3 Преобразование вопросов

После того как обучение завершено, система может преобразовывать вопросы, которых не было в тренировочном наборе. Алгоритм преобразования вопросов QASM [17] сопоставляет вопросу преобразованный запрос, последовательно вычисляя контекст запроса и применяя наиболее вероятный (в соответствии с распределением, заданным в Θ) оператор к запросу, снова пересчитывая контекст и выбирая оператор, и так до тех пор пока в какой-то момент этим оператором не становится *Identity* (или другой оператор, не изменяющий данный запрос). Таким образом, на выходе система выдаст запрос q_s, где

$$q_k = \mathcal{O}_{T(\mathcal{C}(q_{k-1}))}(q_{k-1}),$$

$k \in 1:s$, а q_0-исходный вопрос.

4 Модификации QASM

Кроме описанного в предудыщей секции оригинального алгоритма QASM мы рассматривали два альтернативных подхода, работающих в рамках той же модели.

4.1 Оптимальный QASM (oQASM)

Эта модификация QASM предназначена для оценки максимально достижимого результата при использовании модели QASM. Предполагается, что этот алгоритм всегда возвращает наилучшее преобразование вопроса, которое можно построить при заданом наборе атомарных операторов.

Практическая реализация этого алгоритма была основана на полном переборе всех возможных преобразований для каждого из вопросов тестового набора и выборе дающего наилучший результат.

Отметим, что эта оценка не является верхним пределом в общем случае, т.к. весьма вероятно, что можно добиться лучшего результата при использовании какого-либо другого набора атомарных операторов.

4.2 Множественный QASM (mQASM)

Оригинальный QASM генерирует только один преобразованный запрос по входному вопросу. Однако наши эксперименты показали, что в силу нерегулярности данных в Интернет, даже для очень похожих запросов (неразличимых с точки зрения обученной модели) наилучший результат могут обеспечивать разные преобразования. Например, такими запросами являются вопросы: "`Кто был лауреатом Нобелевской премии мира в 1975 году?`" и "`Кто был лауреатом Нобелевской премии мира в 1979 году?`".

Одной из важнейших причин этого является плохо предсказуемая селективность конкретного запроса. Выбранное алгоритмом преобразование может давать отличные результаты на одном вопросе и, в то же время, приводить к получению запроса нулевой селективности для другого запроса с очень похожими свойствами. Возможен и обратный эффект — слишком неточный (из-за высокой селективности) запрос.

По этой причине естественно рассмотреть такое расширение QASM, которое вместо одного наиболее полезного запроса выбирает наиболее полезное подмножество запросов из множества всех возможных преобразований исходного вопроса. Однако решение этой оптимизационной задачи в общем случае представляется весьма сложным из-за невозможности хорошо предсказать "пересечение" разных запросов. Поэтому мы воспользовались эвристическим предположением, которое заключается в том, что наиболее полезное подмножество запросов — это подмножество всех преобразований с предсказанной полезностью, превышающей некоторое пороговое значение. На этой эвристике основывается алгоритм mQASM.

Построенные алгоритмом mQASM запросы упорядочиваются по убыванию предсказанной селективности (см. раздел 5.4) и последовательно выполняются. Первым поисковой системе отправляется наиболее строгий, из еще неотправленных, запрос. При этом на основании количества найденных по запросу документов оценивается качество поиска[6]. Если приемлемый уровень качества поиска достигнут — то есть найдено достаточное количество документов — до того как все запросы отправлены, то оставшиеся запросы можно уже не выполнять.

Более формально, mQASM генерирует по вопросу q, используя матрицу Θ, построенную на этапе обучения, все возможные запросы $\hat{q}$, полученные последовательным применением атомарных операторов к вопросу, для которых *вероятность выбора* — $P(\hat{q})$ — превышает некоторый порог γ:

$$P(\hat{q}) := \max \prod_{k=0}^{r} p(\mathcal{O}_{s_k}|\mathcal{C}(q_{k-1})) \geq \gamma \qquad (1)$$

где $q_0 = q$, $q_k = \mathcal{O}_{s_k}(q_{k-1})$ и $\hat{q} = q_{s_r}$. Максимум берется по всем последовательностям атомарных операторов $(\mathcal{O}_{s_1}, \ldots, \mathcal{O}_{s_r})$, композиция которых в применении к исходному вопросу q дает запрос $\hat{q}$ (может существовать более одной такой последовательности). $P(q)$ — это и есть предсказанная полезность запроса.

Сгенерированные запросы образуют множество $\hat{Q} = \{\hat{q}|P(\hat{q}) \geq \gamma\}$. Для каждого $\hat{q} \in \hat{Q}$ вычисляется оценка его значимости $w_{\hat{q}}$, которая определяет порядок выполнения запросов. Запросы выполняются в порядке уменьшения значимости, до тех пор пока они не заканчиваются или не собрано достаточное количество документов (обозначим его N_{suff}).

[6] Подобная идея используется в системе Falcon [13], где в зависимости от количества найденных по запросу документов, запрос может быть ослаблен (если их слишком мало) или усилен (если их слишком много) и повторно выполнен.

Результаты a_i выполнения запросов объединяются в единое множество (максимальное число учитываемых ответов на каждый из запросов ограничено сверху той же константой N_{suff}) и упорядочиваются по весу w_{a_i}, вычисляемому как:

$$w_{a_i} = \frac{N_{\text{suff}} - rank_{a_i} + 1}{N_{\text{suff}}} * w_{\hat{q}}$$

где a_i — это один из результатов ответа на запрос $\hat{q}$; $rank_{a_i}$ — порядковый номер a_i в списке ответов на запрос $\hat{q}$ (то есть ранг, присвоенный документу a_i поисковой системой по запросу $\hat{q}$). Если один и тот же результат был получен по нескольким запросам, то в качестве его веса в итоговом объединенном наборе берется наибольший из его весов.

Таким образом, вес документа тем больше, чем ближе документ к началу итогового списка и чем выше оценка значимости запроса, по которому он был получен.

5 Прототип системы

Для проведения экспериментальной оценки алгоритмов мы реализовали прототип системы фактографического поиска для Веб. В качестве базовой поисковой системы использовался Яндекс[7]. Часть описываемых экспериментов мы провели также и с использованием Google[8].

5.1 Атомарные операторы

Мы рассматривали следующие атомарные операторы преобразования запросов:

- Оператор *Identity*, сопоставляющий запросу его самого.
- Несколько операторов удаления слов: удаления стоп-слов, вопросительных слов и операторы удаления слов с частотой, превышающей определенный уровень.
- Операторы склейки: между соседними словами запроса вставляется оператор синтаксиса запросов Яндекс "/n", запрещающий Яндекс возвращать документы, в которых слова из запроса находятся на расстоянии более n слов друг от друга.
- Оператор отмены морфологического анализа: перед каждым словом запроса ставится восклицательный знак, запрещающий Яндекс возвращать документы, в которых данное слово присутствует только в других морфологических формах.

Вероятно, что операторы замены слов на синонимы или обобщающие понятия (а также добавления слов), подобные используемым в [17], могли бы повысить качество поиска, но реализация этих операторов требует использования качественных русскоязычных словарей синонимов или русскоязычного тезауруса.

Отметим, что мы рассматривали специфичные для языка запросов Яндекс операторы для расширения набора операторов, и как оказалось, применение этих операторов положительно сказывается на качестве работы. Однако сами рассматриваемые алгоритмы не привязаны к какому-либо конкретному языку запросов или поисковой системе.

5.2 Свойства запросов

В [16] показывается, что определенные свойства вопросов естественного языка, а именно: тип вопроса (ЛИЧНОСТЬ, МЕСТО и т.п.), число слов в нем и число имен собственных, влияют на способность поисковых систем отвечать на них, и вопросы с одинаковыми значениями этих свойств можно обрабатывать сходным образом.

В нашем прототипе были использованы следующие свойства запросов: тип вопроса, число слов в запросе, число имен собственных, индикаторы применения операторов склейки и отмены морфологического анализа.

5.3 Оценка качества поиска

Для оценки качества ответа на фактографический запрос часто используются следующие метрики:

- **MRR** (Mean Reciprocal Rank) [19]:
  Значение MRR для одного запроса q равно:

  $$MRR(q) = r^{-1},$$

  где r - ранг первого документа, содержащего правильный ответ на вопрос, возвращенного системой среди первых пяти ($r = 0$, если среди первых пяти нет содержащего правильный ответ).

  Т.е. $MRR(q)$ может принимать одно из шести значений: (1, 0.5, 0.33(3), 0.25, 0.2, 0), в зависимости от того, на какой позиции среди первых пяти возвращен документ с правильным ответом.

  Эта метрика несколько лет использовалась для оценки качества фактографического поиска на конференции TREC.

- **TRDR** (Total Reciprocal Document Rank) [17]:
  Эта метрика вычисляется как:

  $$TRDR(q) = \sum_{i=1}^{n_{\text{corr}}} r_i^{-1}$$

  где n_{corr} - число документов, содержащих правильный ответ, среди первых N_{eval}, возвращенных поисковой системой по запросу q; r_i - ранг i-го документа, содержащего правильный ответ.

[7] http://www.yandex.ru
[8] http://www.google.com

Для оценки общего качества поиска по метрикам MRR и TRDR вычисляются их средние значения по всем вопросам тестового набора.

Алгоритмы QASM и mQASM при обучении используют функцию оценки реального качества запроса $F(q)$ (см. раздел 3.2). При выборе метрики, используемой в качестве $F(q)$, мы руководствовались следующими соображениями.

При использовании MRR исходят из того, что для пользователя очень существенно, чтобы правильный ответ был первым в списке документов, поэтому MRR выше, когда система возвращает только один правильный ответ, но на первом месте, чем когда все ответы, кроме первого, в списке результатов правильные. Но в случае, когда результаты поиска документов по сформулированным системой на этапе преобразования запросам передаются другой компоненте системы для выделения ответов, количество документов, содержащих правильный ответ, может иметь важное значение, поскольку многие системы при выделении ответов пользуются избыточностью и склонны выбирать ответы, чаще встречающиеся в результатах поиска. Поэтому метрика TRDR, отличающаяся от MRR тем, что при ее вычислении учитывается не только первый правильный ответ, но и последующие, лучше подходит для оценки качества преобразования запросов, тогда как MRR лучше подходит для оценки качества поиска на выходе полноценной системы фактографического поиска, выделяющей ответы из найденных документов. В частности, проведенные нами эксперименты показали 5%-ное ухудшение качества поиска по метрике MRR и 8.4%-ное по метрике TRDR при использовании при обучении MRR по сравнению с TRDR.

Поэтому для оценки реального качества запросов мы использовали метрику TRDR:

$$F(q) = TRDR(q).$$

5.4 Оценка селективности запроса

Оценка относительной селективности запроса, построенного алгоритмом mQASM, в нашем прототипе выполнялась следующим образом.

Каждому атомарному оператору был сопоставлен некоторый коэффициент селективности s_j, больший или меньший 1. Оператор *Identity*, примененный к запросу, не изменяет его селективность (поэтому коэффициент селективности оператора *Identity* равен 1), операторы удаления увеличивают, остальные уменьшают.

Вес запроса $\hat{q}$ определялся по формуле:

$$w_{\hat{q}} = \prod_j s_j^{-1}$$

где s_j - коэффициенты селективности атомарных операторов, последовательным применением которых к исходному вопросу был получен запрос $\hat{q}$.

Как оказалось, оценка реальных значений коэффициентов селективности операторов слишком трудоемка из-за некоторых особенностей Яндекс[9], поэтому в прототипе использовались эвристические значения этих коэффициентов:

- 1 для оператора *Identity*,
- от 1.05 до 2 для разных операторов удаления слов,
- 0.7 и 0.8 для двух операторов склейки и 0.8 для оператора отмены морфологического анализа.

5.5 Оценка значимости запроса

Алгоритм mQASM (см. раздел 4.2) использует оценки значимости $w_{\hat{q}}$ запроса $\hat{q}$ для определения порядка выполнения запросов, кроме того оценки значимости запросов влияют на итоговые веса результатов поиска.

Мы рассматривали несколько разных вариантов взвешивания запросов:

- **Равные веса**
  Всем запросам присваивается один и тот же вес (равный 1).
- **Оценка вероятности выбора**
  В этом случае вес $w_{\hat{q}}$ запроса $\hat{q}$ считается равным значению $P(\hat{q})$, определяемому формулой 1.
- **Оценка селективности**
  В этом случае весу запроса $\hat{q}$ присваивалась оценка селективности $\hat{q}$.

В каждом из этих вариантов веса нормализовались, так чтобы наибольший вес запроса был равен 1.

6 Экспериментальный анализ

Целью экспериментального анализа являлось изучение поведения алгоритмов QASM и mQASM в сравнении с максимально достижимым результатом, для определения факторов, от которых зависит оптимальный выбор.

6.1 Набор данных

Для обучения системы использовался набор из 60 пар (вопрос,ответ) типа ЛИЧНОСТЬ, из которых 30 были получены из журнала запросов Яндекс, и 30 были придуманы искусственно.

Общая эффективность оценивалась на наборе из 40 вопросов того же типа (все 40 были получены из журнала запросов Яндекс). Наборы вопросов для обучения и для оценки не пересекались.

Решение ограничиться одним типом было обусловлено сложностью задачи создания качественных тренировочного и тестового наборов вопросов. Других принципиальных ограничений, препятствующих оценке других типов фактографических вопросов, в рамках описываемого прототипа нет.

[9] А именно, из-за того, что Яндекс различает несколько уровней соответствия найденных документов запросу: строгих и нестрогих.

Подход	MRR	TRDR	Ответы
Яндекс	0.436	0.938	31 (77.5%)
QASM	0.498 (+14.2%)	0.992 (+5.8%)	29 (72.5%)
mQASM	0.519 (+19.0%)	1.155 (+23.1%)	33 (82.5%)
oQASM	0.678 (+55.3%)	1.457 (+55.2%)	35 (87.5%)

Таблица 2: Общее качество поиска

Отметим, что для запросов других типов вероятно изменение наблюдаемых закономерностей.

6.2 Критерии оценки качества поиска

Для оценки качества поиска использовались три метрики:

- метрика MRR,
- метрика TRDR,
- число вопросов, на которые системе удалось найти правильный ответ.

Оценка выполнялась автоматически: документ, возвращенный системой, засчитывался как правильный ответ, если он содержал соответствие одному из нескольких регулярных выражений, заданных заранее для каждого из вопросов тестового и тренировочного наборов.

При вычислении всех метрик на наличие правильного ответа проверялись только первые 20 возвращенных системой документов (т.е. $N_{\mathrm{eval}} = 20$).

6.3 Общее качество поиска

В таблице 2 приведены результаты оценки общего качества поиска, полученные при использовании алгоритмов QASM, mQASM и oQASM на исходном наборе данных (т.е. при одном из разбиений набора вопросов на вопросы для обучения и оценки).

В ней также для сравнения приведены оценки качества поиска, полученные при выполнении как запросов Яндекс немодифицированных вопросов тестового набора.

Как видно из таблицы, максимально достижимый результат (oQASM) по обеим метрикам MRR и TRDR более чем на 50% превышает результат, полученный при выполнении непреобразованных вопросов. Это подтверждает гипотезу о том, что использование подобных преобразований запросов может значительно повысить качество поиска. Однако результаты QASM и mQASM значительно отстают от максимально достижимого.

QASM превосходит Яндекс по метрикам MRR и TRDR, но проигрывает по числу найденных правильных ответов. Это объясняется тем, что QASM имеет склонность выбирать для вопроса слишком строгое преобразование, которое хорошо подошло для каких-то вопросов тренировочного набора с такими же свойствами. И если по преобразованному запросу находятся документы, то среди них документы, содержащие правильные ответы, имеют высокий ранг (а отсюда и высокий MRR/TRDR). Но в некоторых случаях множество документов, удовлетворяющих слишком строгому преобразованному запросу, пусто. Яндекс же почти всегда возвращает непустое множество ответов, но искомые документы не всегда имеют высокий ранг.

Модифицированный алгоритм, благодаря использованию кроме строгих формулировок запросов и менее строгих, решает проблему QASM и демонстрирует лучшее качество, по сравнению с Яндекс и QASM по всем метрикам.

Однако текущие результаты mQASM пока также значительно уступают максимально достижимым: угадать наилучшее преобразование алгоритму удалось только для 19-ти вопросов тестового набора. Вероятными причинами этого являются: нерегулярность данных в Интернет, недостаточная обученность модели (слишком маленький набор вопросов для обучения) и недостаточно хорошее описание запросов с помощью используемого набора свойств, что не позволяет адекватно обучиться различиям между запросами.

Отметим также, что даже при полном переборе всех возможных преобразований и выборе наилучшего, система смогла найти документы, содержащие правильные ответы, только для 35 вопросов из 40. Этот факт может быть объяснен тем, что документы, содержащие правильные ответы на неотвеченные вопросы, отсутствуют в проиндексированной Яндексом части Интернет, или же им по каким-то причинам присваивается слишком низкий ранг и никакое преобразование вопроса не помогает им оказаться среди первых N_{suff} (в наших экспериментах $N_{\mathrm{suff}} = 20$) документов при ранжировании Яндексом результатов поиска.

6.4 Схемы взвешивания запросов

В таблице 3 представлены результаты экспериментов по сравнению схем взвешивания запросов для модифицированного QASM, описанных в разделе 5.5.

Как и ожидалось, использование равных весов приводит к худшему результату: равноправие запросов обуславливает "зашумление" итогового набора документов при слиянии результатов этих запросов.

То, что оценка селективности запроса оказывается лучше оценки вероятности выбора запроса можно объяснить тем, что наиболее вероятный запрос не всегда является наилучшим: иногда он слишком строг для вопроса (и тогда по нему не находятся никакие документы), иногда слишком

Веса запросов	MRR	TRDR
Равные	0.479	1.059
Вероятность выбора	0.485(+1.2%)	1.134(+7.0%)
Оценка селективности	0.519(+8.4%)	1.155(+9.0%)

Таблица 3: Выбор схемы взвешивания запросов для mQASM.

Без какого оператора?	MRR	TRDR	Ответы
Со всеми операторами	0.519	1.155	33
отмены морфологического разбора	0.437 (-15.8%)	1.001 (-13.3%)	30
склейки	0.485 (-6.6%)	1.093 (-5.3%)	32
удаления	0.396 (-23.7%)	0.967 (-16.2%)	32

Таблица 4: Оценка важности операторов (mQASM)

ослаблен (тогда находится большое число неподходящих). Именно в тех случаях, когда он слишком ослаблен, присваивание ему большего веса, чем наилучшему, и сказывается отрицательно на качестве поиска в схеме взвешивания, основанной на оценке вероятности выбора, поскольку при этом большой вес в итоговом списке могут получить многие неподходящие документы.

6.5 Важность операторов и свойств

Для того чтобы понять, насколько выбор множества используемых операторов и свойств вопроса влияет на качество получаемых результатов, мы поставили две группы экспериментов.

В обоих случаях мы повторяли базовый эксперимент, результаты которого обсуждались в разделе 6.2, но варьировали множества используемых операторов и свойств вопроса.

В первом случае мы повторяли экперимент, поочередно исключая из модели один из операторов. Как видно из результатов в таблице 4, наиболее заметное влияние на результат оказывают операторы удаления и морфологического разбора, причем отмена последнего сказывается на количестве найденных ответов, а не только на их относительном расположении.

Во второй группе экспериментов мы поочередно исключали из модели отдельные свойства, характеризующие вопрос (см. таблицу 5). Наиболее заметное падение эффективности наблюдалось при исключении информации о числе имен собственных.

6.6 Анализ устойчивости

Результаты оценки общего качества поиска, подобные представленным в таблице 2, зависят от используемых наборов вопросов для обучения и оценки, и прежде чем делать выводы, важно оценить стабильность этих результатов.

Вне зависимости от подхода абсолютное качество поиска сильно варьируется в зависимости от набора данных, поэтому усреднение абсолютных величин оценок качества поиска не позволяет сделать осмысленные выводы о стабильности результатов. Вместо этого мы оценивали стабильность выводов о превосходстве каждого из подходов над другими.

Традиционно выводы о превосходстве системы поиска A над системой B на заданном наборе данных делаются на основе измеренных оценок качества поиска по некоторому набору информационных потребностей. [1] При этом разница, которая не превышает определенный *уровень значимости*, считается несущественной, т.е. считается, что ни одна из систем не превосходит другую, и засчитывается ничья.

В этой работе оценивалась стабильность результатов в зависимости от выбранного разбиения множества вопросов на наборы для обучения и оценки. Для этого вопросы набора данных случайным образом разделялись на набор для обучения и набор для оценки в том же соотношении 60/40. Всего было построено 40 случайных разбиений, для каждого из которых были полностью выполнены этапы обучения и оценки.

Результаты экспериментов представлены в таблице 6: в каждой ячейке таблицы стоят разделенные двоеточием количества разбиений, на которых метод, указанный в заголовке строки, соответственно, превзошел, уступил или показал ничейный результат по сравнению с методом, указанным в заголовке столбца.

Как видно из таблицы, множественный QASM в 36 случаях из 40 превзошел результат Яндекс по оценке MRR, а в остальных 4-х вывод о превосходстве сделать невозможно, так как разница результатов не превышает уровня значимости.

QASM часто проигрывает Яндекс вне зависимости от выбранной меры оценки. Тем самым, можно констатировать, что несмотря на позитивный эффект в некоторых случаях, QASM не дает стабильного улучшения и может привести к заметному ухудшению результата.

Множественный QASM ведет себя заметно лучше — он всегда выигрывает у оригинального QASM. Кроме того, он заметно выигрывает у Яндекс по TRDR, хотя по MRR и числу ответов выигрыш не столь значителен. Другими словами, множественный QASM в этих экспериментах стабильно улучшал качество итогового ранжирования.

Без какого свойства?	MRR	TRDR	Ответы
Со всеми свойствами	0.519	1.155	33
числа имен собственных	0.420 (-19.0%)	0.886 (-23.2%)	27
числа слов	0.457 (-11.9%)	1.056 (-8.5%)	32
индикаторов	0.445 (-14.3%)	1.025 (-11.2%)	32

Таблица 5: Оценка важности свойств (для mQASM)

	MRR		TRDR		Ответы	
	Яндекс	QASM	Яндекс	QASM	Яндекс	QASM
QASM	1:30:9	-	2:29:9	-	0:40:0	-
mQASM	17:3:20	36:0:4	37:0:3	40:0:0	5:1:35	40:0:0

Таблица 6: Стабильность выводов о превосходстве (уровень значимости 5%)

Отметим, что хотя подобный анализ стабильности и повышает обоснованность наблюдений, но его результаты могут зависеть от параметров модели (таких как наборы операторов или свойств) и используемого набора данных. Мы планируем в дальнейшем исследовать эти зависимости.

7 Заключение

Огромный объем Веб делает ее весьма привлекательной коллекцией для поиска ответов на фактографические запросы. При обработке таких запросов в контексте Веб важную роль играет качество преобразования вопросов на естественном языке в запросы поисковой системы общего назначения.

В этой работе исследовались возможности использования статистических подходов к преобразованию фактографических запросов на основе алгоритма QASM и его модификаций.

Проведенный экспериментальный анализ показал, что применение преобразований, допускаемых моделью QASM, может значительно повысить качество результатов. Оригинальный алгоритм QASM в наших экспериментах показал нестабильные результаты, но его модификация mQASM вела себя значительно более стабильно.

Основной целью наших исследований является определение и характеризация факторов, влияющих на итоговую эффективность преобразования запросов. Некоторые результаты приведены в этой статье, но эти вопросы конечно же требуют дальнейшего более тщательного и масштабного исследования.

Список литературы

[1] И. Кураленок, И. Некрестьянов. Оценка систем текстового поиска. *Программирование*, 28(4):226–242, 2002.

[2] И. Некрестьянов, Н. Пантелеева. Системы текстового поиска для Веб. *Программирование*, 28(4):207–225, 2002.

[3] Яндекс. Вечные вопросы. *http://www.yandex.ru/skazki/skazka104.html*, Документ был доступен 01.10.2003.

[4] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *Proc. of the WWW-10*, pages 169–178, 2001.

[5] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[6] C.L.A. Clarke, G.V. Cormack, D.I.E. Kisman, and T.R. Lynam. Question answering by passage selection. In *Proc. of the TREC-9*, pages 673–683, 2001.

[7] C.L.A. Clarke, G.V. Cormack, M. Laszlo, T.R. Lynam, and E.L. Terra. The impact of corpus size on question answering performance. In *Proc. of the ACM SIGIR'02*, 2002.

[8] C.L.A. Clarke, G.V. Cormack, T.R. Lynam, C.M. Li, and G.L. McLearn. Web reinforced question answering. In *Proc. of the TREC-10*, pages 673–679, 2001.

[9] C.L.A. Clarke, G.V.Cormack, and T.R. Lynam. Exploiting redundancy in question answering. In *Proc. of the ACM SIGIR'01*, pages 358–365, 2001.

[10] A.P. Dempster, N.M. Laird, and D.B.Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.

[11] Susan Dumais, Michelle Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *Proc. of the ACM SIGIR'02*, pages 291–298, 2002.

[12] Eric Glover, Gary Flake, Steve Lawrence, William P. Birmingham, Andries Kruger, C. Lee Giles, and David Pennock. Improving category specific Web search by learning query

modifications. In *Proc. of the SAINT'01*, pages 23–31, 2001.

[13] Sanda M. Harabagiu, Marius A. Pasca, and Steven J. Mairoano. Experiments with open-domain textual question answering. In *Proc. of the COLIN-2000*, pages 292–298, 2000.

[14] Lieming Huang, Matthias Hemmje, and Erich J. Neuhold. ADMIRE: An adaptive data model for meta search engines. In *Proc. of the WWW-9*, 2000.

[15] Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the Web. In *Proc. of the WWW-10*, pages 150–161, 2001.

[16] Dragomir Radev, Kelsey Libner, and Weiguo Fan. Getting answers to natural language questions on the Web. *Journal of the American Society for Information Science and Technology*, 5(53):359–364, 2002.

[17] Dragomir R. Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Weiguo Fan, and John Prager. Mining the Web for answers to natural language questions. In *Proc. of the ACM CIKM 2001*, pages 143–150, 2001.

[18] M.M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *Proc. of the TREC-10*, 2001.

[19] Ellen M. Voorhees and Dawn M. Tice. The TREC-8 question answering track evaluation. In *Proc. of the TREC-8*, pages 83–105, 2000.

[20] Zhiping Zheng. AnswerBus question answering system. In *Proc. of the HLT 2002*, 2002.

Statistical Query Transformations for Question Answering in the Web

Ilya Boyandin, Igor Nekrestyanov

Abstract

We consider the problem of query transformation for question answering in the Web. The goal of such transformations is to construct a query for a traditional Web search engine given a factual natural language question so that the first several documents returned by the search engine by this query contain the correct answer to the original question.

In this paper we analyse a statistical query transformation algorithms based on the QASM [17] model, and evaluate the search quality of these algorithms on a corpus of correct Russian-language questions from the log of the Yandex search engine.