Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра информатики

Преобразование запросов для предметно-независимого фактографического поиска в Интернет

Дипломная работа
Бояндина Ильи Игоревича

Научный руководитель: .....
/к.ф-м.н. Некрестьянов И.С./

Рецензент: .....
/д.ф-м.н., профессор Новиков Б.А./

"Допустить к защите",
заведующий кафедрой
информатики: .....
/д.ф-м.н., профессор Косовский Н.К./

Санкт-Петербург
2003

Оглавление

Глава 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 специальная дорожка посвященная экспериментальной оценке систем фактографического поиска [19].

Огромный объем доступной в Интернет текстовой информации дает системам фактографического поиска, осуществляющим поиск в документах Интернет, потенциальную возможность находить ответы на очень многие фактографические вопросы. Информация в сети постоянно обновляется, пополняется новыми данными, в отличие от информации в закрытых коллекциях документов. В то же время, данные в Интернет, из-за отсутствия централизованного контроля над публикацией, обладают рядом особенностей, осложняющих решение задачи поиска: неструктурированностью, разнородностью, противоречивостью. В Интернет также наблюдается избыточность, повторяемость данных: ответ на один и тот же вопрос, по-разному сформулированный, может содержаться в десятках документах сети. Многие исследования показывают, что избыточность информации в Интернет может быть использована для повышения эффективности фактографического поиска [11, 9]. Экспериментальные данные для некоторых систем демонстрируют повышение эффективности при поиске в Интернет по сравнению с результатами, достигнутыми теми же системами при поске в коллекциях меньшего размера [7, 8].

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

1. **Предварительный**

[1] за 20 декабря 1999 года

На этом этапе обычно выполняется классификация вопроса, на основании результатов которой, формулируется запрос и определяются возможные формы ответов.

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 убедится 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 по запросу "`Когда родился Набоков?`" только один документ из пяти первых содержит правильный ответ[2]. Качество преобразования вопроса в запрос поисковой системы в значительной мере определяет общую эффективность системы фактографического поиска. Действительно, если ни один документ содержащий искомый ответ не будет найден

[2]Описанный эксперимент с Яндекс проводился 15 марта 2003

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].

При проведении этой работы были поставлены следующие цели:

- изучить современные методы фактографического поиска, и в частности, подходы к преобразованию вопросов в запросы поисковых систем общего назначения,
- реализовать прототип системы фактографического поиска на русском языке, использующей один из современных алгоритмов преобразования запросов,
- экспериментально оценить эффективность алгоритма и выявить его слабые стороны,
- оценить максимальную эффективность, которой можно добиться с помощью алгоритма, построенного на подобных принципах.

Работа имеет следующую структуру: в главе 2 представлен обзор близких работ; в главе 3 описаны алгоритм QASM и его предлагаемая модификация; некоторые детали реализации прототипа системы представлены в главе 4, а результаты экспериментов представлены в главе 5.

[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.

Глава 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

Вопрос	Тип вопроса
Как звали 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 2.1: Примеры типов вопросов

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 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 2.1), который опреде-

ляется 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[1].

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 узкоспециализированные, 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.

[1] Отметим, что такой подход значительно повышает нагрузку на последующие шаги обработки фактографического запроса. Хотя в этом случае система получает больше документов с правильными ответами и это повышает вероятность его удачно извлечения, система также получает кучу мусора, который может ввести ее в заблуждение.

Шаблон вопроса	Шаблоны запросов поисковой системы
What is a <тело вопроса>	“<тело вопроса>” AND “is a” “<тело вопроса>” AND “refers to” ...
Who is a <тело вопроса>	“<тело вопроса>” AND “is a” “<тело вопроса>” AND “usually is” ...

Таблица 2.2: Примеры преобразований, которым обучается система Tritus

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 (см. таблицу 2.2). Затем система оценивает эффективность 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.

В статье [17] описывается алгоритм QASM, который обучается преобразованиям вопросов на наборе вопросов и ответов. Этот алгоритм активно используется в этой работе и детально описан в следующей главе.

Глава 3

Описание алгоритмов

3.1 Алгоритм QASM

Вероятностный алгоритм QASM[1] 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[2]. 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.

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 этому классу.

[1] Question Answering using Statistical Models

[2] Термины *вопрос* и *запрос* различаются: *вопросом* будем называть только исходный вопрос, заданный пользователем; а *запросом* может быть, как исходный вопрос, так и вопрос, к которому были применены какие-либо операции преобразования.

3.1.1 EM-алгоритм

В основе алгоритма обучения QASM лежит алгоритм максимизации ожидания (Expectation Maximization) - итеративный алгоритм нахождения оценок максимального правдоподобия. Этот алгоритм 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 повторяет следующие два шага до тех пор пока не достигает локального максимума:

1. оценивает неизвестные параметры, используя все доступные ему данные,

2. используя полученные оценки, модифицирует собственную модель.

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

3.1.2 Обучение

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

Пусть $\mathcal{A}_1, \ldots, \mathcal{A}_l$ — фиксированный набор функций (*свойств запросов*), сопоставляющих запросу целое число. 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 *реальной эффективности* запроса, ко-

торая может использовать информацию о найденных по этому запросу документах.

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 эффективному 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 для каждого контекста и модифицируется в процессе обучения. Построенная матрица Θ и есть результат обучения алгоритма.

Псевдокод алгоритма обучения QASM приведен в листинге 3.1.1. Алгоритм обучения последовательно выполняется для каждого из вопросов тренировочного набора на одной и той же матрице Θ. Шаги алгоритма, повторяющиеся, пока не выполнены условия завершения, такие:

1. Применить каждый из операторов к запросу, оценивая эффективность получаемых запросов (зная правильный ответ на исходный вопрос). Если наибольшей эффективностью обладает оператор *Identity*, то обработка текущего запроса завершена. Иначе — выбрать по распределению вероятностей, заданному в Θ для контекста запроса, оператор, который будет применен к запросу.

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

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

   Если изменение матрицы Θ на этой итерации не превышает заданного порога ($\delta(\Theta) < \varepsilon$), то обработка текущего запроса завершена. Иначе цикл повторяется с первого шага.

3.1.3 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*

Листинг 3.1.1 Алгоритм обучения QASM

```
/*
  T = { набор вопросов и ответов для обучения }
  O = { набор всех атомарных операторов преобразования }
  C(q) - контекст запроса q
  op(q) - означает применение оператора op к запросу q
*/

foreach (q, a) in T do
begin
  repeat
    f:=array[sizeof(O)]
    foreach op in O do
      f[op]:=Оценить_реальную_эффективность(op(q), a)
    if (максимальную реальную эффективность дает Identity) then
        break
    op:=Выбрать_по_Theta_наиболее_вероятный_оператор_для(C(q))
    q:=op(q)
    Внести_изменения_в_Theta(C(q), f)
  until δ(Θ) < ε
end
```

(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.

Листинг 3.1.2 Алгоритм преобразования запросов QASM

```
/*
  q0 - входной вопрос
  C(q) - контекст запроса q
  op(q) - означает применение оператора op к запросу q
*/

  q:=q0
  repeat
    op:=Выбрать_по_Theta_наиболее_вероятный_оператор_для(C(q))
    if (op <> Identity) and (op(q) <> q) then
        q:=op(q)
  until (op = Identity) or (op(q) = q)
```

3.2 Модифицированный QASM

Описанный выше алгоритм преобразования вопросов генерирует только один преобразованный запрос по входному вопросу.

Однако, практическое применение подобного подхода к задаче фактографического поиска в Интернет демонстрирует, 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?".

Разные преобразования в применении к одному и тому же вопросу могут давать запросы различной селективности: более строгие или менее строгие. По более строгим запросам поисковая система находит меньшее число результатов, но этим результатам можно доверять в большей степени. Если выбранная алгоритмом последовательность операций преобразует вопрос в слишком строгий запрос, то дав отличный результат на одном вопросе, она может привести к нулевой селективности на другом вопросе с такими же свойствами из-за нерегулярности данных в Интернет. С другой стороны, последовательность операций, преобразующая вопрос в запрос с высокой селективностью, хорошо сработав для одного вопроса, может дать слишком много неподходящих результатов для другого.

Эти рассуждения приводят к тому, что вместо одного наиболее вероятного запроса по входному вопросу имеет смысл строить несколько наиболее вероятных запросов. Построенные запросы можно выполнять последовательно, начиная с наиболее строгого, оценивая качество поиска на основании количества найденных по запросу документов. Когда приемлемый уровень качества поиска достигнут, то есть найдено достаточное количество документов, оставшиеся запросы можно уже не выполнять.

Предлагаемая модификация алгоритма QASM основана на этой идее.

Подобный подход 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.

Модифицированный алгоритм 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:

$$P(\hat{q}) := \max \prod_{k=0}^{r} p(\mathcal{O}_{s_k}|\mathcal{C}(q_{k-1})) \geq \gamma \tag{3.1}$$

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).

Псевдокод подпрограммы преобразования запросов модифицированного алгоритма приведен в листинге 3.2.1.

Сгенерированные запросы образуют множество $\hat{Q} = \{\hat{q}|P(\hat{q}) \geq \gamma\}$. Для каждого $\hat{q} \in \hat{Q}$ вычисляется[3] 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}).

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 на вес документа в итоговом объединенном наборе влияет значимость запроса, с помощью которого он был найден, и ранг документа в соответствующей выдаче.

[3] В действительности оценка значимости вычисляется уже при генерации преобразованных запросов (алгоритм, приведенный в листинге 3.2.1, несколько упрощен).

Листинг 3.2.1 Подпрограмма преобразований запросов в modQASM

```
/*
  O = { набор всех атомарных операторов преобразования }
  C(q) - контекст запроса q
  Theta(op, c) - функция, возвращающая элементы матрицы Θ
  op(q) - означает применение оператора op к запросу q
*/

function Найти_преобразования_для(q, P)
begin
  R:=∅
  foreach op in O do
  begin
    P_:=P * Theta(op, C(q))
    if (P_ >= γ) then
    begin
      if (op = Identity) or (op(q) = q) then
        R:=R ∪ {q}
      else
        R:=R ∪ Найти_преобразования_для(op(q), P_)
      end
    end
  end
  return R
end
// Пример вызова:
Найти_преобразования_для("Кто был первым космонавтом?", 1.0)
```

Глава 4

Прототип системы

Для проведения экспериментальной оценки алгоритмов автором был реализован прототип системы фактографического поиска в Интернет, использующий Яндекс[1] в качестве базовой поисковой системы.

В этой главе описываются концептуальные решения, которые были приняты при реализации системы.

4.1 Атомарные операторы

В реализованном прототипе использовались следующие атомарные операторы преобразования запросов (см. раздел 3.1.2):

- 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 $/n$, запрещающий Яндекс возвращать документы, в которых слова из запроса находятся на расстоянии больше n слов.

[1] http://www.yandex.ru

- Оператор отмены морфологического анализа: перед каждым словом запроса ставится восклицательный знак, запрещающий Яндекс возвращать документы, в которых данное слово присутствует, но только в другой морфологической форме.

Некоторые из этих операторов используют возможности, специфичные для языка запросов Яндекс, и как оказалось, применение этих операторов положительно сказывается на эффективности. Но нужно заметить, что сами рассматриваемые алгоритмы не привязаны к какому-либо конкретному языку запросов или поисковой системе.

Весьма вероятно, что операторы замены слов на синонимы или обобщающие понятия (а также операторы добавления слов), подобные используемым в [17] могли бы повысить эффективность системы, но реализация этих операторов для системы фактографического поиска на русском языке требует использования русскоязычных словарей синонимов или русскоязычного тезауруса.

4.2 Свойства запросов

В [16] показывается, что определенные свойства вопросов естественного языка, а именно: тип вопроса (см. таблицу 2.1), 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 реализованном прототипе были использованы следующие свойства запросов: тип вопроса, число слов в запросе, число имен собственных, индикаторы применения операторов склейки и отмены морфологического анализа.

4.3 Оценка реальной эффективности запроса

Алгоритм QASM при обучении опирается на функцию оценки реальной эффективности запроса $F(q)$ (см. раздел 3.1.2). В реализованном прототипе для вычисления $F(q)$ использовалась метрика TRDR (Total Reciprocal Document Rank) [17]:

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

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

Например, если среди найденных поисковой системой по запросу q документов второй, третий и пятый содержали правильный ответ, то

$$F(q) = \frac{1}{2} + \frac{1}{3} + \frac{1}{5}$$

Метрика TRDR похожа на другую стандартную метрику — MRR [19], которая несколько лет использовалась для оценки эффективности систем фактографического поиска на конференции TREC.

Однако, TRDR более чувствительна к различиям запросов, поскольку учитывает не только первый правильный ответ, возвращенный системой, но и следующие за ним правильные ответы, поэтому ее имеет смысл использовать при обучении.

В частности, проведенные автором эксперименты показали 5%-ное падение эффективности по метрике MRR и 8.4%-ное по метрике TRDR при использовании при обучении MRR вместо TRDR.

4.4 Оценка значимости запроса

Предложенная модификация QASM (см. раздел 3.2) использует оценки значимости $w_{\hat{q}}$ запроса $\hat{q}$ для определения порядка выполнения запро-

сов, кроме того оценки значимости запросов влияют на итоговые веса результатов поиска.

В этой работе рассматривались несколько разных вариантов взвешивания запросов для модифицированного QASM:

- **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 3.1.

- **Оценка селективности**

  Каждому атомарному оператору 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 $\hat{q}$ определяется 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[2], поэтому в прототипе использовались эвристические значения этих коэффициентов:

[2] А именно, из-за того, что Яндекс различает несколько уровней соответствия найденных документов запросу: строгих и нестрогих, и в возвращаемом Яндекс наборе документов могут быть документы, в разной степени соответствующие запросу. Из-за этого для оценки селективности запроса приходится загружать с сайта Яндекс большое количество страниц с результатами поиска, что требует слишком больших ресурсов для запросов с высокой селективностью.

– 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.

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

Глава 5

Экспериментальный анализ

Целями экспериментов являлись выяснение максимальной эффективности, которой можно достичь с помощью алгоритмов преобразования запросов, подобных QASM, оценка эффективности алгоритма QASM и модифицированного алгоритма, оценка стабильности модифицированного алгоритма, а также вкладов разных составляющих модифицированного алгоритма в итоговую эффективность.

Далее в этой главе описываются использованный набор данных, критерии оценки качества поиска по преобразованным запросам и проведенные эксперименты.

5.1 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 (см. Приложение А).

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 во-

просов. Других принципиальных ограничений, препятствующих оценке других типов фактографических вопросов в рамках описываемого прототипа, нет. Отметим, однако, что для запросов других типов возможно значительное изменение наблюдаемых закономерностей.

5.2 Критерии оценки качества

Для оценки результатов использовались уже упомянутая ранее в разделе 4.3 метрика TRDR и метрика MRR (Mean Reciprocal Rank) [19].

Среднее значение MRR по всем вопросам тестового набора вычисляется по формуле:

$$MRR = \frac{1}{n}\sum_{i=1}^{n} RR_i$$

где

$$RR_i = rank_i^{-1}$$

$rank_i$ - ранг первого документа, содержащего правильный ответ на i-й вопрос тестового набора, возвращенного системой среди первых пяти ($RR_i = 0$, если среди первых пяти нет содержащего правильный ответ); n - число вопросов тестового набора.

При оценке систем фактографического поиска с использованием 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 эффективности на выходе полноценной системы фактографического поиска, выделяющей ответы из найденных документов.

При оценке эффективности в этой работе вычислялись значения обеих метрик. При вычислении оценок по метрике TRDR на наличие правильного ответа проверялись первые 20 возвращенных документов, т.е. $N_{\mathrm{TRDR}} = 20$.

5.3 Эксперименты

В этом разделе описываются проведенные эксперименты и обсуждаются полученные результаты.

5.3.1 Максимально достижимый результат

Для оценки максимально достижимого результата, который можно получить с помощью преобразований вопросов при фиксированном наборе атомарных операторов, был выполнен полный перебор всех возможных преобразований для каждого из вопросов тестового набора и взят наилучший результат.

Вычисленная таким способом оценка эффективности поиска по обеим метрикам MRR и TRDR превышает более чем на 50% оценку, полученную при выполнении непреобразованных вопросов. Этот результат - красноречивое свидетельство в пользу того, что с помощью преобразований запросов можно добиться значительного повышения эффективности.

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.

Подход	MRR	TRDR	Ответы
Яндекс	0.436	0.938	31 (77.5%)
QASM	0.498 (+14.2%)	0.992 (+5.8%)	29 (72.5%)
modQASM	0.519 (+19.0%)	1.155 (+23.1%)	33 (82.5%)
Макс. достиж. эфф-сть	0.678 (+55.3%)	1.457 (+55.2%)	35 (87.5%)

Таблица 5.1: Общая эффективность

Отметим, однако, что даже при полном переборе всех возможных преобразований и выборе наилучшего система смогла найти документы содержащие правильные ответы только на 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}[1] документов при ранжировании Яндексом результатов поиска.

5.3.2 Общая эффективность

В таблице 5.1 приведены результаты оценки общей эффективности прототипа, полученные с использованием алгоритмов QASM и modQASM (модифицированного QASM), и для сравнения - максимально достижимый результат. В ней также для сравнения приведены оценки эффективности, полученные при выполнении в качестве запросов Яндекс немодифицированных вопросов тестового набора.

Кроме значений MRR и TRDR для каждого из подходов в таблице приведено число правильных ответов, которые удалось найти системе (ответ засчитывался, если среди одного из первых 20-ти документов, возвращенных системой в итоговом списке, был хотя бы один, содержащий правильный ответ).

[1] В реализованном прототипе $N_{\text{suff}} = 20$

QASM превосходит Яндекс при выбранном разбиении набора вопросов 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 заметно лучшую эффективность по сравнению с Яндекс и QASM по всем метрикам.

Однако, текущие результаты modQASM 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.

5.3.3 Схемы взвешивания запросов

В таблице 5.2 представлены результаты экспериментов по сравнению схем взвешивания запросов для модифицированного QASM, описанных в разделе 4.4.

Как и ожидалось, схемы взвешивания, использующие оценки вероятности выбора запросов и оценки селективности, превосходят схему с

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 5.2: modQASM: Схемы взвешивания запросов

равными весами: при использовании равных весов документы, которым Яндекс присвоил высокий ранг, найденные как по более вероятным запросам, так и по менее вероятным, как по более строгим, так и по менее строгим, имеют одинаковые шансы получить большой вес в итоговом множестве результатов, вследствие чего документы, содержащие правильные ответы, оказавшиеся в начале итогового списка, "разбавляются" неподходящими документами.

При этом схема взвешивания, основанная на оценке селективностей, превосходит основанную на оценке вероятности выбора запросов. Это можно объяснить тем, что наиболее вероятный запрос, к сожалению, не всегда является наилучшим: иногда он слишком строг для вопроса (и тогда по нему не находятся никакие документы), иногда слишком ослаблен (тогда находится большое число неподходящих). Именно в тех случаях, когда он слишком ослаблен, присваивание ему большего веса, чем наилучшему, и сказывается отрицательно на эффективности в схеме взвешивания, основанной на оценке вероятности выбора, поскольку при этом большой вес в итоговом списке могут получить многие неподходящие документы.

Эти результаты подтверждают перспективность выбранного подхода к модификации QASM, использующего селективности запросов для оценки их значимости.

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

Таблица 5.3: modQASM: Вклад операторов

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

Таблица 5.4: modQASM: Важность свойств

5.3.4 Важность операторов и свойств

В таблицах 5.3 и 5.4 представлены результаты экспериментов по оценке вклада атомарных операторов и свойств запросов, использованных в реализованном прототипе, в общую эффективность системы. В этих экспериментах и обучение, и оценка эффективности проводились на тех же наборах вопросов, но без использования соответствующих операторов или свойств. Как видно из таблицы 5.3, самый большой вклад в общую эффективность вносит оператор удаления слов, но в то же время, оператор, запрещающий Яндекс морфологический разбор слов, сильно влияет на число вопросов, на которые системе удалось найти ответ. Среди всех использованных свойств наиболее важное, без сомнений — число имен собственных в вопросе.

5.3.5 Выбор параметра γ

В модифицированном алгоритме важную роль играют вероятности выбора запросов (см. раздел 3.2). Запрос используется при поиске документов, если вероятность его выбора превосходит порог γ.

Для выбора наилучшего значения γ в этой работе была проведена серия экспериментов: общая эффективность прототипа оценивалась при различных значениях γ. Полученная зависимость эффективности от выбора значения γ иллюстрируется графиками на рисунке 5.1. Графики были построены по результатам, полученным для около 70 различных значений γ (остальные точки на графиках были построены линейной интерполяцией) с уменьшением шага при значениях γ, близких к нулю.

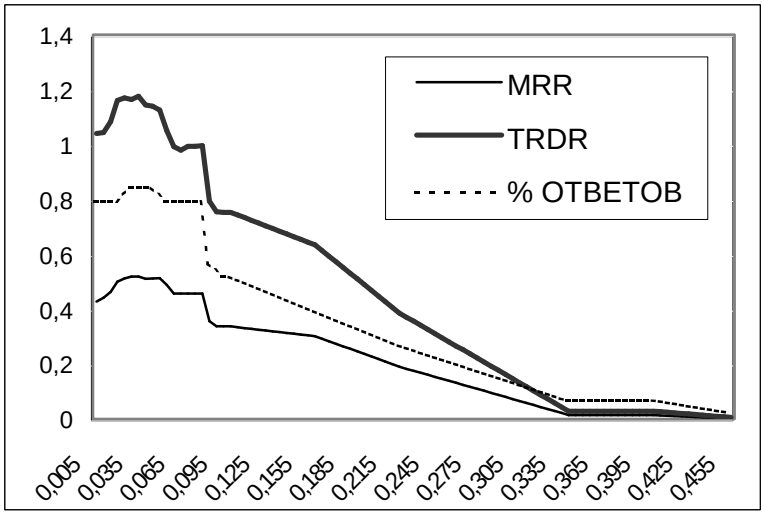

Рис. 5.1: Зависимость эффективности от γ

Из графиков видно, что параметр γ заметно влияет на эффективность по всем метрикам, и что существует некоторый диапазон значений, для которых эффективность наиболее высока.

При росте γ и приближении к 1 эффективность стремится к нулю, поскольку при этом для каждого из вопросов тестового набора количество его преобразований, для которых $P(\hat{q}) \geq \gamma$ уменьшается и при каком-то

$\gamma_0 < 1$ достигает нуля. При стремлении γ к нулю эффективность застывает на определенном уровне, меньшем максимального. Это объясняется тем, что при очень маленьких γ все возможные запросы, которые могут быть получены из вопросов с помощью атомарных преобразований, генерируются алгоритмом. При $\gamma = 0$ алгоритм зацикливается.

Максимальное значение эффективности по всем метрикам в проведенных экспериментах достигалось при $\gamma = 0.035$. Это значение и было использовано при проведении всех остальных экспериментов (в том числе и при оценке общей эффективности).

5.3.6 Анализ устойчивости

Результаты оценки общей эффективности, представленные в таблице 5.1, зависят от используемых наборов вопросов для обучения и оценки, поэтому важно оценить стабильность этих результатов. Абсолютная эффективность каждого из подходов сильно варьируется в зависимости от набора данных, поэтому усреднение абсолютных величин оценок эффективности не позволяет сделать осмысленные выводы о стабильности результатов. Вместо этого стоит оценивать 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 эффективности поиска по некоторому набору информационных потребностей, при этом разница, которая не превышает определенный *уровень значимости*[2] не учитывается [1].

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 сгенерировано 40 разбиений, для каждого из которых

[2]Обычно уровень значимости составляет 5% от абсолютного значения оценок. В описываемых экспериментах использовалось именно такое значение.

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

Таблица 5.5: Стабильность выводов о превосходстве

были полностью выполнены этапы обучения и оценки.

Χ Результаты экспериментов представлены в таблице 5.5: в каждой ячейке таблицы стоят разделенные двоеточием количества разбиений, на которых метод, указанный в заголовке строки, соответственно, превзошел, проиграл и дал результат, отличающийся на величину, меньшую уровня значимости, от результата, полученного с использованием метода, стоящего в заголовке столбца.

Как видно из таблицы, подход, использующий QASM, в реализованном прототипе проигрывает модифицированному QASM и Яндекс для большинства разбиений. Следовательно, можно констатировать, что алгоритм QASM не дает стабильной приемлемой эффективности при выбранном наборе операторов и свойств.

В то же время, модифицированный QASM выигрывает у QASM по всем метрикам, стабильно выигрывает у Яндекс по TRDR, и чаще всего выигрывает или дает результат, незначительно отличающийся от Яндекс, по метрике MRR. По числу найденных ответов модифицированный QASM для большинства разбиений дает результат, незначительно отличающийся от Яндекс (это можно объяснить тем, что и при использовании модифицированного алгоритма поиск документов осуществляет тоже Яндекс: modQASM часто находит по преобразованным запросам те же документы, которые могут быть найдены по исходному вопросу, поскольку преобразованные запросы близки к исходному вопросу; но modQASM ранжирует найденные документы более удачно, из-за чего его оценка по TRDR выше). При этом модифицированный QASM чаще

выигрывает у Яндекс по числу ответов, чем проигрывает.

Таким образом, можно говорить о том, что подход, использующий модифицированный QASM, при выбранным наборе операторов и свойств демонстрирует стабильное превосходство над QASM и Яндекс.

Глава 6

Заключение

Огромный объем Интернет делает его весьма привлекательной коллекцией для поиска ответов на фактографические вопросы. При обработке таких вопросов в контексте Интернет важным фактором общей эффективности фактографического поиска является качество преобразования вопросов на естественном языке в запросы поисковой системы общего назначения.

В этой работе исследовалась возможность повышения качества поиска документов, содержащих вероятные ответы на фактографические вопросы, с помощью таких преобразований. В работе получены следующие результаты:

- изучены современные подходы к созданию систем фактографического поиска,
- реализован алгоритм преобразования запросов QASM,
- оценена максимально достижимая эффективность алгоритмов преобразования запросов, основанных на используемых QASM принципах, и наборе операторов, частично описывающих возможности синтаксиса запросов Яндекс,
- проанализирована работа QASM на основе тестового набора данных и выявлены слабые места этого алгоритма,

- предложена модификация QASM, в которой исправлены некоторые его недостатки, и экспериментально проверено превосходство нового алгоритма над QASM,
- проведена экспериментальная оценка стабильности и параметров модели.

Полученные результаты свидетельствуют о перспективности использования алгоритмов, обучающихся преобразованиям запросов, для фактографического поиска и мотивируют проведение дальнейших исследований для установления причин отставания эффективности предложенного алгоритма от максимально достижимого результата и способов уменьшения этого отставания.

В будущем автор планирует провести более тщательный анализ влияния параметров алгоритма и характеристик вопросов, используемых для обучения и оценки, на общую эффективность, а также исследовать возможность создания более эффективных методов преобразования запросов и влияние эффективности преобразований на последующие этапы фактографического поиска.

Приложение А

Набор данных

В приложении приведен список вопросов, использованных при проведенных экспериментах. Вопросы с 1 по 69 были взяты из журнала Яндекс, а с 70 по 100 придуманы автором. В экспериментах по оценке общей эффективности для обучения использовались вопросы с 41 по 100, а для оценки с 1 по 40. Для каждого из вопросов набора автором вручную были написаны по одному или нескольким регулярных выражений, с помощью которых результаты поиска проверялись на содержание правильных ответов.

1. Какой итальянский поэт был лауреатом Нобелевской премии в 1959 году?
2. Кто был самым первым президентом Украины?
3. При каком правителе к России была присоеденена территория Финляндии?
4. Кто доказал необходимость витаминов для человеческого организма?
5. В честь кого назван аэропорт в Мюнхене?
6. Кто играл Гарри Поттера?

7. Памятник кому герой Савелия Крамарова в “Джентельменах удачи” назвал “мужиком в пиджаке”?

8. Кто создал литературный образ детектива лорда Питера Вимси?

9. Кто автор известной книги о Гарри Поттере?

10. Кто вычистил Авгиевы конюшни?

11. Кто открыл Америку?

12. Кто сражался с немейским львом?

13. Какой древнегреческий бог соответствует индуистскому божеству Кама?

14. Кто явился инициатором возобновления Олимпийских игр и основателем современного олимпийского движения?

15. Как зовут героя Гете, доктора, который ухаживал за Маргаритой?

16. Какая настоящая фамилия Анны Ахматовой?

17. Кто из композиторов написал цикл оперных произведений под общим заголовком “Кольцо Нибелунгов”?

18. Кто сказал “Красота спасет мир”?

19. Кто впервые выдвинул идею перехода на летнее время?

20. Кто основал город Львов?

21. Какой русский царь подавил восстание Болотникова?

22. Какой бог жил во дворце Валгалла?

23. Как звали императора “Священной Римской империи” по прозвищу Барбаросса?

24. Кто исполнял роль эльфа Леголаса в фильме “Властелин колец”?

25. Кто завершил первое кругосветное путешествие Магеллана после его гибели на Филлипинах?

26. Кто прорубил окно в Европу?

27. Как звали врага Вильгельма Телля, принудившего его стрелять в яблоко, лежащее на голове собственного сына?

28. Как звали олениху - подругу Бемби?

29. Как зовут гавайскую богиню вулканов?

30. Какой русский художник принимал участие в разработке буденновки?

31. Кому принадлежит фраза “После нас хоть потоп”?

32. Как звали богиню, покровительницу мчртвых, в египетской мифологии?

33. Кто из киевских князей учредил “Шапку Мономаха”?

34. Кто помог напечататься двенадцатилетнему поэту Михалкову?

35. Кто был дедушкой верховного бога Зевса?

36. Кто играл темнокожую певицу в фильме “Мы из джаза”?

37. Кто убил Кеннеди?

38. Кого традиционно считают одним из создателей храма Василия Блаженного в Москве?

39. Как зовут бога подземного мира в мифах ацтеков?

40. Кто ввел впервые термин числитель и знаменатель?

41. Кто отец Ады Лавлейс, первого в мире программиста?

42. Кто открыл Саргассово море?

43. Кто создал образ мисс Марпл?

44. Кто возглавил, согласно мифологии, поход греков против Трои?

45. Под каким именем больше известна Софья Васильевна Корвин-Круковская?

46. Как звали музу комедии?

47. Как звали мать Александра Македонского?

48. Как звали возлюбленную Наполеона, ставшую его первой женой?

49. Как зовут солиста группы Rammstein?

50. Кто закончил свое стихотворение утверждением "Истина в вине"?

51. Кто из российских хирургов впервые провел операцию под наркозом на поле боя?

52. Кто изобрел книгопечатание в России?

53. Кто написал сказку “Синяя борода”?

54. Кто первым выиграл чемпионат “Формула-1”?

55. Именно под этим именем мы знаем Долорес, горячо любимую Гумбертом?

56. Именно эта женщина придумала праздник 8 марта?

57. Имя американской танцовщицы, пассии Сергея Есенина?

58. Имя гнома, предводителя отряда, в книге Р. Толкиена “Хоббит”?

59. Из произведений какого писателя физики заимствовали слово “кварк”?

60. Кто архитектор Николаевского дворца в Петербурге?

61. Кто первый написал фантастический роман “Люди как боги”?

62. Кто написал “Пиноккио”?

63. Кто из русских князей впервые принял титул царя?

64. Как звали злую сестру доктора Айболита в сказке Корнея Чуковского?

65. Кем был построен памятник Пушкину в Москве?

66. Кто автор повести, по которой поставлен знаменитый фильм Альфреда Хичкока “Птицы”?

67. Как зовут единственную женщину-фараона?

68. Кого играет Джулианна Мур в фильме “Часы”?

69. Кто автор биографии Дэвида Копперфильда?

70. Кто был первым президентом СССР?

71. Кто был руководителем крестьянского восстания 1773 года в России?

72. Кто был изобретателем вертолета?

73. Кто был первым космонавтом?

74. Кто был открывателем Америки?

75. Как звали изобретателя лампочки?

76. Кто был назван журналом Таймс человеком столетия в 1999 году?

77. Кто был лауреатом Нобелевской премии мира в 1975 году?

78. Кто был лауреатом Нобелевской премии мира в 1979 году?

79. Кто был лауреатом Нобелевской премии мира в 1989 году?

80. Как звали генерального секретаря ООН, которому запретили въезд в США?

81. Кто был автором первого учебника по органической химии?

82. Как зовут автора книги “Архипелаг Гулаг”?

83. Кто был режиссером фильма “Солярис”?

84. Как зовут создателя Linux?

85. Как зовут создателя Линукс?

86. Как звали жену Пушкина?

87. Как зовут библейского героя, предавшего Иисуса?

88. Как зовут писателя, которому принадлежит фраза “красота спасет мир”?

89. Как звали австрийского психолога, создателя логотерапии?

90. Как звали создателя логотерапии?

91. Как зовут актера, исполнившего роль Гитлера в фильме “Молох”?

92. Как зовут актера, исполнившего роль Ленина в фильме “Телец”?

93. Как звали музу, покровительницу трагедии в греческой мифологии?

94. Как зовут главного редактора радио “Эхо Москвы”?

95. Назовите имя известного российского музыканта, играющего на альпийском горне?

96. Назовите имя исполнителя на альпийском горне?

97. Назовите имя лидера группы Jethro Tull?

98. Назовите имя героя Сервантеса, который боролся с мельницей?

99. Назовите имя библейского героя, сила которого заключалась в волосах?

100. Как звали жену Станиславского?

Литература

[1] Кураленок И.Е. and Некрестьянов И.С. Оценка систем текстового поиска. *Программирование*, 2002.

[2] Некрестьянов И.С. and Пантелеева Н. Системы текстового поиска для Веб. *Программирование*, 2002.

[3] Яндекс. Вечные вопросы. *http://www.yandex.ru/skazki/skazka104.html*, Документ был доступен 5.03.2003.

[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, May 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 TREC-9*, 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 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 (MultiText experiments for TREC 2001). In *Proc. of TREC-9*, 2001.

[9] C.L.A. Clarke, G.V.Cormack, and T.R. Lynam. Exploiting redundancy in question answering. In *Proc. of ACM SIGIR'01*, 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 ACM SIGIR'02*, 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 SAINT-2001*, 2001.

[13] Sanda M. Harabagiu, Marius A. Pasca, and Steven J. Mairoano. Experiments with open-domain textual question answering. In *Proc. of COLIN-2000*, 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, May 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), 2002.

[17] Dragomir 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 ACM CIKM 2001*, 2001.

[18] M.M. Soubbotin. Patterns of potential answer expressions as clues to the right answers. In *Proc. of TREC-10*, 2002.

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

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