Московский Государственный Университет

имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра системного программирования

Дипломная работа

Исследование и разработка современных методов реализации вопросно-ответных систем

Выполнил:

студент 528 группы

Агаев Нурлан Закирович

Научный руководитель:

к. ф.-м. н. Турдаков Денис Юрьевич

Москва

2013

Оглавление

Аннотация

В данной работе рассматривается предметная область вопросно-ответного поиска, исследуются существующие методы анализа запроса и поиска ответа в системах для английского языка. В результате исследований был построен прототип вопросно-ответной системы для русского языка и были произведены экспериментальные исследования прототипа, в работе описывается архитектура построенной системы и используемые методы.

Введение

В последнее время наблюдается сильный рост объема общедоступной информации. Огромное количество как структурированных, так и неструктурированных данных доступно нам посредством сети Интернет. По причине этого существует проблема поиска и получения необходимой нам информации. Данная задача поиска и обработки информации еще больше усугубляется тем, что информация в Интернет находится в постоянном изменении и обновлении, имеет высокий уровень динамики. В каждый момент времени появляются новые материалы, новые факты. Объем информационных массивов постоянно увеличивается, вследствие этого необходим постоянный учёт информации, что зачастую является невозможным.

В настоящее время для пользователя Интернет доступны системы информационного поиска. Данные системы требуют запроса, сформулированного из ключевых слов, которые соответствуют текстовой информации, нужной для пользователя. Необходимо отметить, что зачастую данные системы ну учитывают порядок слов, их формы и связи между самими словами, то есть рассматривают запрос, просто как небольшой набор слов. В то время как человеку более свойственно формулировать запросы в форме вопроса на естественном языке. В результате работы поисковых систем выдается большое количество ссылок и текстовых фрагментов в порядке релевантности, то есть дальнейший поиск информации должен вести сам пользователь, что может затруднить её восприятие и увеличить время поиска. Когда мы хотим что-то узнать, мы спрашиваем – задаём вопрос, что, в общем, и естественно в процессе познания. Большинство систем по поиску информации, не имеют возможности отвечать на наши вопросы. Для поиска и получения человеку нужно сформировать запрос из ключевых слов и задать его поисковой машине.

Согласно краткой информационной бюллетени департамента маркетинга компании Яндекс [7] каждый день русскоязычные пользователи по оценке компании Яндекс задают более 85 миллионов запросов и общая доля поисковых запросов, сформулированных в виде вопроса, составляет более 3 %. Отчет охватывает все системы, которые базируются на поисковой площадке Яндекс. Вопросительными словами, с которых начинается большинство вопросительных запросов, на июль 2009 являются «какой», «сколько», «кто» и «где». Среди поисковых запросов, заданных как вопрос, больше половины не повторяются. Самая маленькая доля уникальных запросов с вопросительным словом «кто» - 38 %. То есть, эта конструкция подразумевает наибольшее однообразие заданных вопросов.

В последние годы интересы исследователей всё более перемещается в сторону интеллектуального поиска информации. Повысился интерес к разработке интеллектуальных и нетрадиционных механизмов поиска и получения информации. Интернет стал рассматриваться в качестве потенциальной большой базы знаний, для работы с которой требуются специальные инструменты. К задаче поиска информации также относится и поиск ответов на вопросы на естественном языке. Так на ежегодном форуме CLEF (Conference and Labs of the Evaluation Forum) [13] , на котором проводят независимую оценку методов информационного поиска, ориентированных на работу с англоязычными информационными материалами, можно наблюдать постоянный интерес к разделу вопросно-ответного поиска QA@CLEF за последние несколько лет (рисунок 1). Увеличилось и количество проектов таких систем в данной области информационного поиска, области поиска ответа на вопрос на естественном языке.

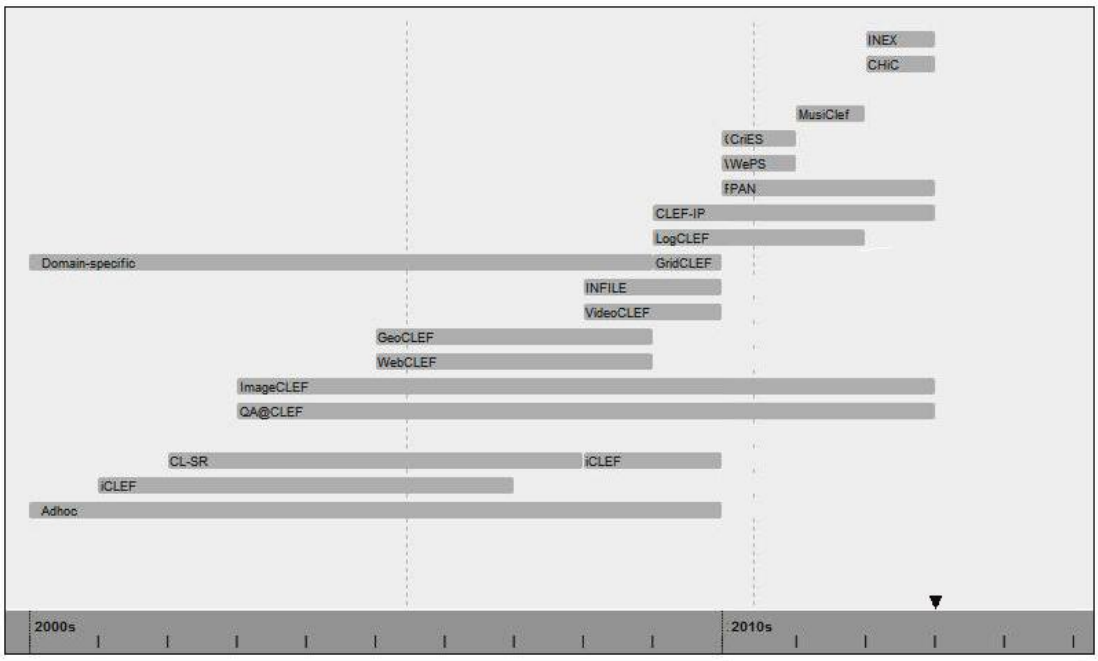

Рисунок 1. Разделы (дорожки) CLEF с 2000 по 2013 годы [13].

Вопросно-ответные системы – это программные комплексы, которые умеют обрабатывать введенные пользователем вопросы на естественном языке и давать на них краткие ответы, состоящие из слов или предложений. Вопросно-ответные системы имеют другую цель по сравнению с традиционными системами информационного поиска. Их задача – найти небольшой фрагмент документа, представляющий точный и краткий ответ на вопрос. Источником информации для таких систем обычно служит большая коллекция текстовых документов, например, общедоступные страницы сети Интернет. Таким образом, вопросно-ответные системы образуют класс интеллектуальных систем информационного поиска.

При разработке и реализации вопросно-ответных систем приходится иметь дело с естественным языком, а именно с фразами и предложениями, сформированные по определенным правилам этого языка. При проектировании таких систем используются относительно новые методы компьютерной лингвистики (англ. «NLP – Natural Language Processing»), требуется применение адекватных лингвистических средств по работе с

естественным языком, при этом результат работы системы существенно зависит от качества их реализации. Известные реализации вопросно-ответных систем демонстрируют невысокие показатели качества, в частности, точности поиска. Например, наилучший результат в дорожке вопросно-ответного поиска в области биомедицины в 2012 показал точность, равную 0.55 [1][13].

В последние годы появилось немало проектов в данном направлении. Причем это проекты, в которых разработаны технологии обработки простых вопросов, ответы на которые состоят из одного слова или небольшого предложения. Эти проекты обходят стороной вопросы более сложного вида, например, вопросы причины, описания объектов и т.д. Авторы большинства создаваемых в настоящее время систем вопросно-ответного поиска ориентируются в основном на английский язык. Для русского языка на момент исследований данной работы существует система AskNet[2], но научные публикации по ней и подробности методов реализации отсутствуют. Типовая вопросно-ответная система состоит из большинства сложных частей, которые предназначены для анализа вопроса и обработки текстовых документов с учетом правил и особенностей естественного языка. Например, при анализе вопроса происходит синтаксический и семантический разбор предложения. Части и подпрограммы анализа разрабатываются обычно независимыми группами и, как уже было отмечено, работают с английским языком. Поэтому для русского языка затруднительно применить готовые модули обработки предложений. Данные факты являются серьезным препятствием для разработчиков, желающих создать системы для русского языка. При анализе вопроса и текстовых документов для извлечения из них ответа используют различные анализаторы языка: синтаксические,

[1] В данном случае точностью является отношение количества правильно отвеченных вопросов к общему количеству вопросов.

[2] http://www.asknet.ru

морфологические, семантические. К сожалению, выбор русскоязычных систем анализа, представленных в свободном доступе довольно скуден.

Любая серьезная система вопросно-ответного поиска должна каким-либо образом производить анализ структуры вопросительного предложения, опираясь на знания о конкретном естественном языке, на котором сформулирован вопрос. Вследствие этого изучение и сравнение решений, рассчитанных на разные языки – достаточно сложная задача. Однако для исследования принципов работы существующих систем, технологий решения задачи вопросно-ответного поиска, позволят сделать вывод о глубине анализа и обработки вопросительных предложений и текстовых документов, что будет использовано при построении системы для русского языка.

1 Постановка задачи

Целью данной работы является исследование методов реализации современных вопросно-ответных систем, а также разработка и построение прототипа вопросно-ответной системы для русского языка, способной выдавать ответы на вопросы об одушевлённых объектах и географических местах. Для достижения данной цели были поставлены следующие задачи:

1. провести исследование предметной области вопросно-ответного поиска и существующих методов реализации вопросно-ответных систем;
2. разработать прототип вопросно-ответной системы для русского языка, способной выдавать ответы на вопросы об одушевлённых объектах и географических местах;
3. провести экспериментальные исследования разработанного прототипа.

2 Обзор предметной области

2.1 Вопросно-ответный поиск

Вопросно-ответный поиск – это вид информационного поиска, при котором на вопрос, заданный пользователем на естественном языке, можно получить краткий ответ. Современные системы информационного поиска позволяют нам получить список целых документов, которые могут содержать интересующую информацию, при этом оставляют пользователю работу по получению нужных данных из документов, упорядоченных по уровню релевантности запросу. Системы вопросно-ответного поиска в сравнении с традиционными поисковыми системами получают вопросительно предложение на естественном языке (на английском, на русском и т.д.), а не набор ключевых слов, и возвращают краткий ответ, а не список документов и ссылок. Например, пользователь вводит следующий вопрос: «Кто является президентом России?» и в качестве ответа получает имя человека, а не список релевантных ссылок на текстовые документы. Таким образом, нахождение ответа на вопрос извлечением небольшого отрывка текста из документа, в котором непосредственно содержится сам ответ, в отличие от информационного поиска совсем другая задача. Ответ на вопрос пользователя должен быть кратким, достоверным и актуальным.

В системах вопросно-ответного поиска активно используются технологии обработки естественных языков (англ. «natural language processing) [1]. Сначала на вход системе подаётся запрос, сформулированный в виде вопросительного предложения на естественном языке. Далее входной запрос обрабатывается, происходит поиск и вывод ответа в виде одного или нескольких слов на естественном языке, а может быть и небольшого фрагмента текста. Поиск ответа может производиться по некоторой информационной текстовой базе. Источником информации для поиска ответа может быть как Интернет, так и локальное хранилище данных. Локальное

хранилище данных может быть в виде коллекции проиндексированных документов. Вопросно-ответные системы отличаются как видом поиска информации, так и тематикой вопросов, которые они могут обрабатывать.

2.2 Виды вопросно-ответных систем

Условно системы вопросно-ответного поиска можно разделить на следующие группы: общие, специализированные (узкого назначения) [1]. Первый вид вопросно-ответных систем ориентирован на обработку любых вопросов или, по крайней мере, большинства их видов. В свою очередь специализированные системы рассчитаны на обработку вопросов узкой тематики, то есть вопросов по определенной предметной области (медицины, искусства и т.д.). Также данные группы могут отличаться и способом поиска информации. В качестве источника информации общие системы в основном используют большой корпус документов или чаще всего сеть Интернет. Специализированные системы могут использовать свою специальную локальную коллекцию документов на тему предметной области. Общие системы в связи с тем, что зачастую они обрабатывают неструктурированную информацию, имеют достаточно сложную внутреннюю структуру и используют различные технологии обработки естественных языков. Далее приводится исследование и рассмотрение существующих методов реализации вопросно-ответных систем с учетом таких характеристик, как способ информационного поиска, наличие и вид информационной базы данных, необходимость привлечения экспертов предметной области, вычислительная сложность работы, архитектурная гибкость системы.

Подходы к реализации и соответственно принципы построения вопросно-ответных систем можно разделить на следующие несколько групп:

- вопросно-ответные системы, базирующиеся на веб-поиске (англ. «web-based question answering system»);

- вопросно-ответные системы с собственной размеченной коллекцией документов;
- вопросно-ответные системы с базой данных, содержащей вопросы и ответы;
- вопросно-ответного системы экспертного типа.

2.2.1 Вопросно-ответные системы, базирующиеся на веб-поиске

Вопросно-ответные системы, базирующиеся на веб-поиске, в качестве источника используют веб-страницы или их фрагменты. При построении данных систем используется результаты систем информационного поиска сети Интернет, то есть в данном случае в архитектуру включена одна из существующих поисковых систем [1][2]. Вопросно-ответная система, получая вопросительное предложение на естественном языке от пользователя, обрабатывает его, генерирует запрос из ключевых слов для поисковой системы. Ключевые слова выбираются исходя из самого вопросительного предложения. После поискового запроса система получает результаты информационного поиска в виде веб-ссылок и фрагментов текста – сниппетов. Сниппет - небольшой отрывок текста из веб-документа результатов работы поисковой системы, который используется в качестве описания ссылки в результатах поиска. Обычно сниппет содержит контекст, в котором встретилось ключевое слово в тексте веб-документа. Далее вопросно-ответная система работает с данными фрагментами веб-документов, используя методы обработки естественных языков, генерирует ответ пользователю. Обычно это методы выделения различных именованных сущностей, дат, чисел и различные алгоритмы выбора фрагмента текста в качестве ответа [5][10][15]. Также могут использоваться и синтаксические, морфологические методы анализа текста[10][15]. На рисунке 2 показана обобщенная архитектура вопросно-ответной системы, базирующейся на

Интернет. Подробнее о работе данного вида вопросно-ответных систем будет рассказано далее в этой работе.

Отметим основные преимущества данных систем:

- нет необходимости хранения и индексирования большого количества текстовой информации;
- использование сторонних компонентов в виде поисковой системы освобождает от решения задач разработки поиска релевантных документов;

Но есть и недостатки данного подхода к разработке вопросно-ответных систем:

- качество информации зависит от результатов работы выбранной поисковой системы, но в настоящее время популярные поисковые системы выдают достаточно точные и релевантные результаты;
- размер получаемых сниппетов зависит от выбранной поисковой системы и не может быть изменён;
- текстовая информация в сниппетах неструктурированная.

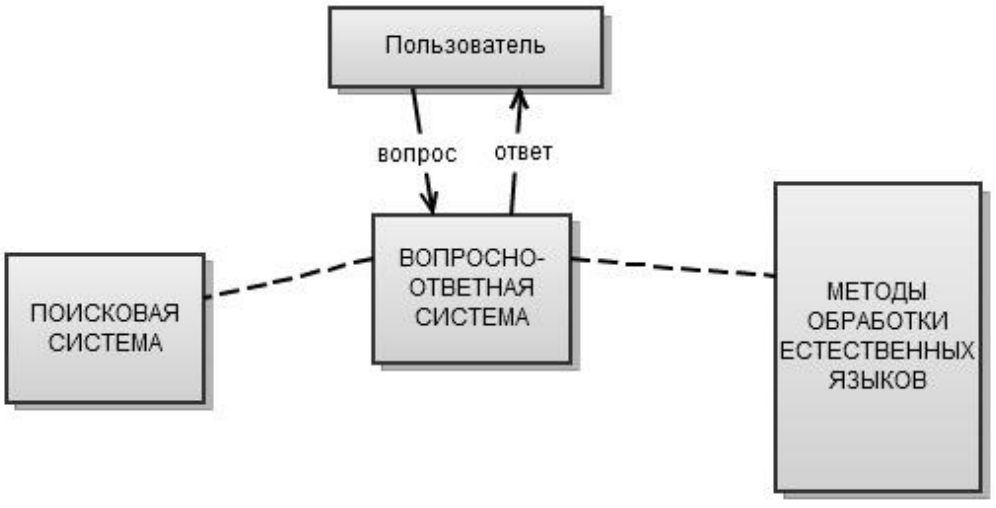

Рисунок 2. Обобщенная архитектура вопросно-ответной системы, базирующейся на Интернет

2.2.2 Экспертные опросно-ответные системы

Другим типом вопросно-ответных систем являются экспертные. Пользователь, которому необходимо получить ответ на вопрос, задает его системе. При этом система может потребовать некоторые пояснения. Далее вопрос на естественном анализируется системой, внутренним алгоритмом, использующим специальную большую базу знаний, строится решение вопроса и выводится ответ. Обычно экспертные системы настроены на работу в рамках некоторой предметной области и имеют узкую специализацию.

Архитектура данных систем предусматривает наличие специальной базы знаний, информация в которой хранится в виде некоторой структуры. В базе знаний содержится вся необходимая информация для принятия решений. Почти любая база знаний должна содержать достоверные данные о выбранной предметной области. Единицами знаний в базе являются факты. Также в базе знаний присутствуют правила, так называемые продукции. Каждая продукция по своей сути - это просто программа из выражения вида «если выражение-1, то выражение-2 », где «выражение» – это факт или комбинация фактов с логическими операциями И/ИЛИ. При помощи последовательности этих элементарных программ определяется набор разрешенных преобразований от начального состояния до окончательного решения поставленной задачи. С помощью продукций система может получать новые данные о предметной области. Причем новые данные тоже будут достоверными. Эти продукции с помощью специальной программы могут добавляться, изменяться и удаляться. Это делается экспертом предметной области. Также и база знаний заполняется экспертом. На рисунке 3 представлена обобщенная архитектура вопросно-ответной системы экспертного типа.

Преимуществами вопросно-ответных систем экспертного типа являются:

- сравнительно высокая скорость работы;

- высокая достоверность ответа;

Недостатками таких систем являются:

- необходимость привлечения экспертов и необходимость поддержки системы;

- необходимость создания объемной базы знаний, содержащей структурированную информацию и различные правила (продукции);

- отсутствие архитектурной гибкости системы; сильная зависимость от структуры фактов (фреймовой модели), которая может быть адекватна для одной определенной предметной области.

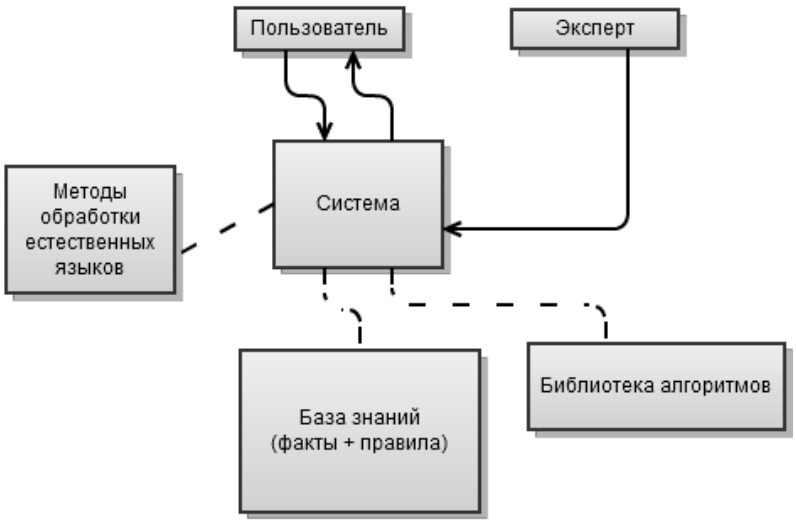

Рисунок 3. Обобщенная архитектура вопросно-ответной системы экспертного типа

Обычно это требует постоянной работы экспертов определенной предметной области, необходимы чёткое понимание потребностей пользователей и очень внимательная проработка модели данных, постоянное расширение и модификация этой модели для новых предметных областей;

- необходимость наличия исходных текстов для извлечения достоверной информации об окружающем мире;

- большая вычислительная и организационная трудоёмкость построения базы фактов; лингвистическая анализ на глубоком семантическом уровне извлечения фактов подвержена большому количеству ошибок, так как может собирать все ошибки лингвистической обработки на предшествующих уровнях: графематическом, морфологическом, синтаксическом и семантическом;

Одним из ярких примеров вопросно-ответной системы экспертного типа является интеллектуальная система WolframAlpha[3]. Отметим то, что данная система не является поисковой интеллектуальной системой, хотя и имеет похожий интерфейс. Алгоритм WolframAlpha основан на обработке естественного языка, на данный момент поддерживает только английский, наличии базы знаний и большого множества алгоритмов вывода информации из неё. WolframAlpha не возвращает при запросе список ссылок или фрагментов страниц, а вычисляет ответ, основываясь на собственной большой базе знаний, большой библиотеке и алгоритмов и NKS-подходе [14]. NKS-подход предусматривает наличие множества простых программ-правил (продукций), для формирования полного ответа. База знаний WolframAlpha содержит структурированную информацию по различным предметным областям: медицине, математике, физике, биологии, астрономии, химии, истории, географии, а также музыки, кинематографии, политики и биографическую информацию об известных людях. Данная система способна переводить численные данные между различными системами счисления, системами измерения, подбирать общую формулу для заданной в запросе численной последовательности, вычислять самые различные математические выражения. Часть программного кода WolframAlpha для реализации данной прикладной задачи написана на

[3] http://www.wolframalpha.com/

языке Mathematica[4] и составляет около 5 миллионов строк, в настоящее время выполняется примерно на 10000 процессорах.

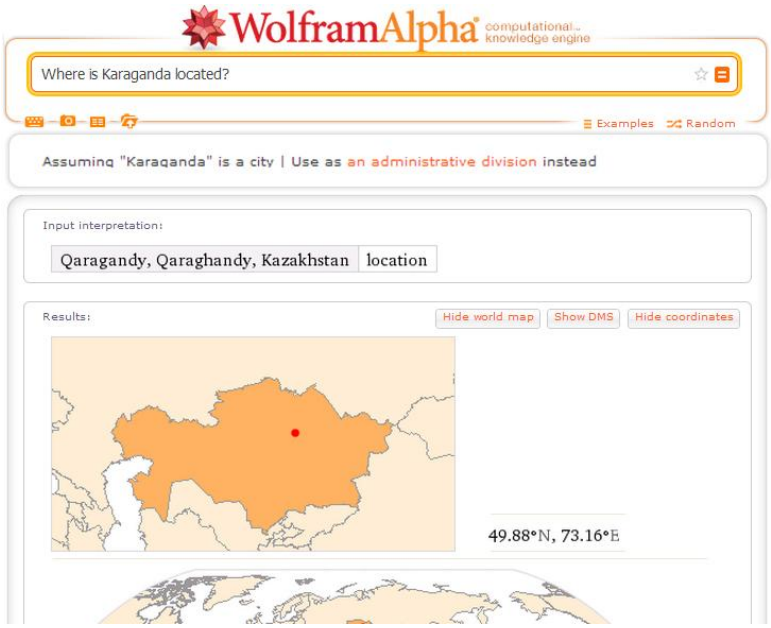

Рисунок 4. Пример работы системы WolframAlpha.

2.2.3 Вопросно-ответные системы с локальной коллекцией вопросов и ответов

Другим подходом к разработке вопросно-ответных систем являются системы с коллекцией вопросов и ответов [1][3]. Такая архитектура предусматривает локальную коллекцию вопросов и соответствующих им ответов или же открытую базу вопросов и ответов, организованную через социальную систему посредством веб-портала, в котором каждый может задать вопрос или ответить на вопрос другого пользователя. Пользователь задает вопрос на естественном языке, затем система производит поиск похожих вопросов в коллекции вопросов и ответов и выдаёт ответ на поставленный вопрос. В открытых системах выдается наиболее похожий вопрос со списком ответов, в начале которого указан ответ, за корректность которого проголосовало наибольшее количество пользователей, если такого

[4] http://en.wikipedia.org/wiki/Mathematica. Mathematica - это интерпретируемый язык функционального программирования.

похожего вопроса не нашлось, то вопрос пользователя заносится в коллекцию вопросов и остаётся открытым для обсуждения и ответа другими пользователями. Данные в такой системе представлены в виде коллекции вопросов с ответами, которая может пополняться другими пользователями. В таких системах база вопросов и ответов может пополняться и автоматически – это происходит поиском по сети Интернет и анализом текстовых документов, в ходе которого могу быть извлечены пары «вопрос-ответ», также система может и автоматически задавать вопросы по тексту после соответствующего анализа.

Достоинствами систем с коллекцией вопросов и ответов можно считать:

- возможность развёрнутых необязательно фактографических ответов;
- возможность обходиться без использования сложных алгоритмов;
- корректность ответов проверяется.

Но в то же время есть и недостатки:

- необходимость организации открытой социальной системы обмена ответами;
- необходимо время для дальнейшего пополнения коллекции вопросов;
- проверка ответов другими пользователями, а следовательно необходимо привлечение пользователей;
- организация объемного хранилища.

Одним из других подходов к реализации вопросно-ответных систем, является применение локальной коллекции индексированных документов [6]. Архитектура таких систем предусматривает модуль поиска по индексу таких документов. Этот поисковый индекс, в отличие от случая классических поисковых систем, дополняется специфическими для вопросно-ответной системы атрибутами. Элементами индекса являются не отдельные слова текста, а объекты лингвистического анализа, например [7][8]:

- именованные сущности;

- элементарные синтаксические связки (пары грамматически связанных слов и др.).

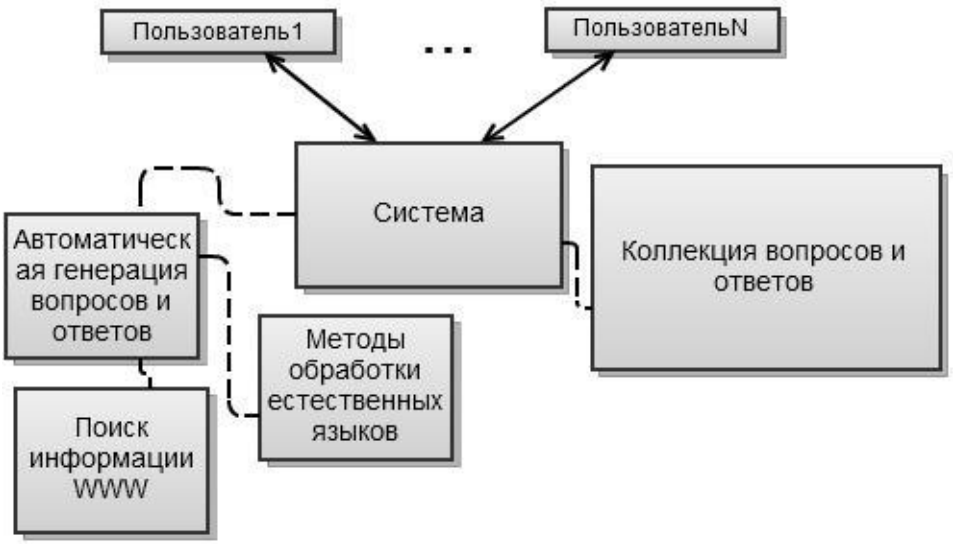

Рисунок 5. Обобщенная архитектура социальной вопросно-ответных систем с коллекцией вопросов и ответов.

При помощи элементов обработки естественных языков система строит индекс: каждый новый документ проходит автоматическую обработку текста на естественном языке, размечаются требуемые системе объекты, затем они добавляются в индекс.

Использование своего специального индекса позволяет преодолеть некоторые недостатки архитектуры, основанной на интернет-поиске.

К достоинствам данного типа систем можно отнести следующие:

- меньшая вычислительная трудоёмкость в момент обработки вопроса пользователя в реальном времени благодаря собственному индексу;

- собственный индекс позволяет организовать наиболее удобный для системы поисковый аппарат.

Недостатками являются:

- невысокая гибкость по сравнению с системами, основанными на веб-поиске: на этапе построения индекса выбирается какая-то определённая

модель представления и индексирования текста. Любые изменения, скорее всего, потребуют перестройку части системы, отвечающей за индексацию и поиск по индексу;

- необходимость индексации текстовой информации, что требует вычислительных затрат. Учитывая то, что документы анализируются целиком, хоть и ответы на вопросы пользователей содержатся в очень редких предложениях, можно сказать, что ресурсы могут расходоваться неэффективно.

Приведенное деление вопросно-ответных систем по архитектуре и принципам разработки является устоявшимся, но есть примеры систем, объединяющих в себе несколько подходов и способов реализации. Например, суперкомпьютерная система IBM Watson проекта DeepQA [9]. IBM Watson – интеллектуальная суперкомпьютерная система, понимающая вопрос на естественном языке и выдающая на них ответ. Система, которая изначально задумывалась для участия в вопросно-ответной телевикторине Jeopardy (аналогом которой в РФ является телешоу «Своя игра»), выиграла в итоге турнир викторины. Система имеет возможность поиска информации по сети Интернет, также есть локальное хранилище данных, которое содержит более 200 миллионов страниц структурированной и неструктурированной информации, полностью включая весь актуальный текст Википедии[5] на английском языке (более 4.2 млн. статей). Технологическая платформа построена на суперкомпьютерном кластере, состоящем из 90 серверов Power7 750 и имеющем 2880 вычислительных ядер и оперативную память объёмом 15 Терабайт. Система создавалась группой лучших исследователей компании IBM в течение более четырёх лет и поддерживается большой командой разработчиков, инженеров, экспертов и аналитиков.

[5] Википе́дия — общедоступная многоязычная универсальная интернет-энциклопедия. Расположена на интернет-сайте http://www.wikipedia.org/.

2.3 Этапы вопросно-ответного поиска

Большая часть существующих проектов в области вопросно-ответного поиска предназначены для английского языка. Если сравнить несколько работ в данной сфере исследований, то можно прийти к выводу, что процесс поиска ответов в текстовых документах можно разделить на следующие этапы: этап анализа вопроса, этап информационного поиска, этап извлечения ответа[1][3][11].

2.3.1 Анализ вопроса

На этапе анализа вопроса происходит ввод пользователем вопроса на естественном языке и дальнейшая обработка. Как правило, современные вопросно-ответные системы способны обрабатывать некоторые предопределённые классы вопросов. Вопросы можно разделить по виду ответа на следующие: фактографические вопросы, вопросы причины и содержания, вопросы мнения [3][8]. Фактографический вопрос – это вопрос о различных сведения без их анализа, обобщения, освещения, ответ на данный вопрос обычно краток. Примерами фактографических вопросов являются вопросы о персонах, о времени, вопросы, требующие ответа «да» или «нет». Примерами вопросов причины являются следующие: «Почему небо голубое?», «Как умер Брюс Ли?». Вопросы причины и мнения требуют логического анализа текста вывода причинно-следственной связи между предложениями и являются самыми сложными в плане нахождения ответа, ответ на такой тип вопроса не является точным и обычно им может являться обычно несколько предложений [8]. Существующие решения в области анализа текста и вывода его логической структуры довольно плохо справляются с данным видом вопросов. Вопросы мнения, предусматривают собой поиск и глубокий анализ блогов и различных сайтов СМИ. Наибольший интерес вызывают фактографические вопросы. Ответ на

фактографический вопрос является точным, кратким и актуальным. Именно на обработку данного вида вопросов направлена разработка современных вопросно-ответных систем. В свою очередь фактографические вопросы можно разделить на следующие:

- требующие ответа «да» или «нет»;
- вопросы о персонах;
- вопросы о географических топонимах;
- вопросы о списках чего-либо;
- вопросы об определениях и т.д.

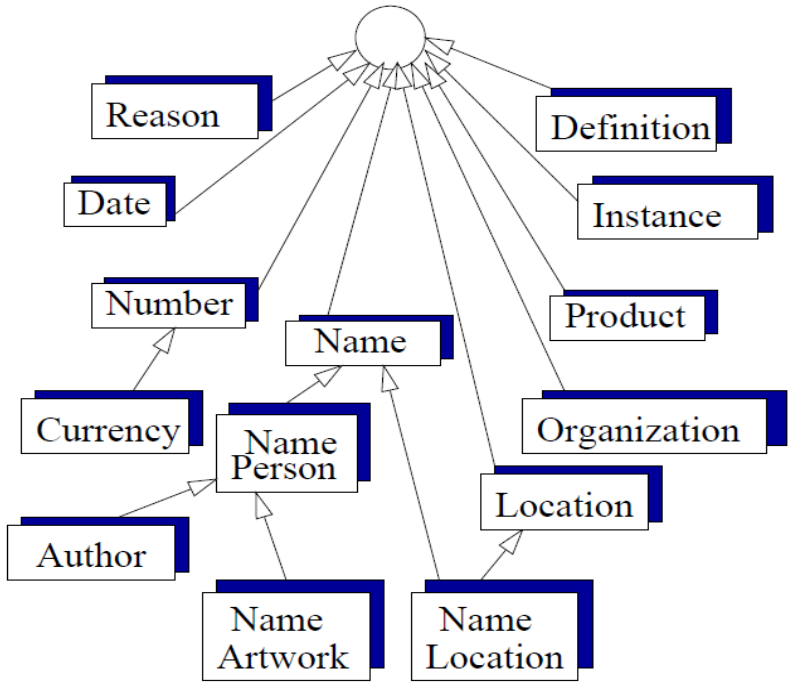

Рисунок 6. Пример иерархии типов ожидаемого ответа

На этапе анализа вопроса изначально происходит выявление класса вопроса, соответствующего типу возможного ответа на заданный вопрос. Задача состоит в определении типа информации, которая интересует пользователя и может выдаться в качестве ответа – типа ожидаемого ответа (англ. «expected answer type») [5]. Тип ожидаемого ответа используется в

дальнейшем для определения стратегии поиска ответа на этапе его извлечения. Если он известен, то при извлечении ответа допускается не рассматривать те предложения, в которых нет словесных единиц, соответствующих данному типу ожидаемого ответа. Многие вопросно-ответные системы имеют встроенную систему поддерживаемых типов ожидаемого ответа, иногда используется иерархическая структура представления в виде таксономии в зависимости от дальнейшей стратегии поиска и извлечения ответа [11]. На рисунке 6 представлен пример таксономического дерева типов ответов, используемого в системе [2]. Разные системы могут иметь разную классификацию. В ранее упоминаемой системе IBM Watson введена таксономия из 2500 типов.

В большинстве случае распознавание типа ожидаемого ответа производится с помощью использования набора регулярных выражений или текстовых шаблонов. При работе системы производится сопоставление вопросительного предложения с заранее построенными шаблонами или регулярными выражениями, соответствующими типам ответов. В таблице 1 приведены примеры регулярных выражений для английского языка, использующиеся в системе [23]. Как правило, данные выражения и шаблоны пишутся вручную, но могут создаваться автоматически по принципам машинного обучения с учителем. Например, с помощью размеченного корпуса вопросов [24]. Система Webclopedia имеет встроенную структуру из более 280 написанных вручную шаблонов, соответствующих 180 типам ответа. На этапе анализа вопроса может производиться синтаксический анализ вопросительного предложения. При классификации вопроса используется синтаксическая информация о частях речи в вопросительном предложении и наличии именованных сущностей. Одним из простейших способов является определение типа вопроса по вопросительным словам, которые содержит запрос к системе: «кто», «что», «какой», «где», «когда» и

т.д. Для английского языка распознавание типа ответа может быть произведено с помощью анализа вопросительных слов предложения (wh-words) [2] [6].

Таблица 1. Примеры классов вопросов и шаблонов

Класс вопроса	Примеры шаблонов
name	/(W|w)hat(wa| i|\')s the name/
pers-def	/[Ww]ho(wa| i|\')s [A-Z][a-z]+/
number	/[Hh]ow (much|many) /
thing-def	/[Ww]hat(wa| i|\')s an? /, / (was|is|are|were) a kind of what/
date	/[Ww]hen /, /[Ww](hat|hich) year /
location	/[Ww]here(\'s)? /, / is near what /
capital	/[Ww]hat is the capital /, /[Ww]hat is .+\'s capital/
date-death	/[Ww]hen .* die/, /[Ww](hat|hich) year .* die/
date-birth	/[Ww]hen .* born/, /[Ww](hat|hich) year .* born/

Определение типа ожидаемого ответа сужает множество возможных ответов и определяет дальнейшую стратегию поиска возможного ответа [11].

Недостатками подхода с использованием шаблонов являются:

- невозможность покрыть большую часть реальных вопросов пользователей. Вопросы подбираются таким образом, чтобы обработать конкретный набор тестовых заданий, при этом выйти за пределы этого покрытия «неудобным вопросом» достаточно легко;
- соответствие между шаблонами и ожидаемыми типами ответов не так прямолинейно. Например, слово «кто» в вопросе может подразумевать не только персону, но и организацию или народ (к примеру, ответом на вопрос «Кто сочиняет народные песни?» будет слово «народ»).

Следующей задачей, решаемой на этапе анализа вопроса, является поиск ключевых слов для формирования запроса к поисковому модулю вопросно-ответной системы. Ключевые слова – слова, используемые в качестве запроса для информационного поиска. Данные слова будут для поиска и выбора документов. Точный вид запроса зависит от реализации системы. При использовании интернет-поисковой машины (например, Yahoo, Bing) в запрос можно включить каждое слово предложения, поскольку современные поисковые машины устойчивы к стоп-словам. При создании запроса из вопросительного предложения могут быть удалены вопросительные слова (например: «кто», «что», «какой»), так как они обычно не встречаются в ответах. Удалением этих слов можно повысить полноту[6] (англ. «recall») результатов информационного поиска. Для этого можно опять же использовать регулярные выражения.

Одним из способов формулирования поискового запроса является также перефразирование вопросительного предложения в неполное утвердительное предложение [6]. Например, для вопроса «Кто был первым человеком в космосе?» результатом перефразирования будет следующее предложение: «Первым человеком в космосе был». Далее представлены примеры правил перефразирования вопрос в вопросно-ответной системе Lin для английского языка [5]:

Wh-word did A verb B ? → A verb+ed B

Where is A ? → A is located in

Модуль анализа и обработки вопроса очень важен в современных системах, если он не будет работать корректно, это создаст проблемы для остальных модулей.

[6] Отношение числа найденных ответов на вопросы к общему числу вопросов.

После того, как определены ключевые слова, то есть сформулирован поисковый запрос, работа системы переходит на следующий этап – этап информационного поиска.

2.3.2 Информационный поиск

На данном этапе происходит получение релевантных поисковому запросу текстовых фрагментов, которые возможно содержат ответ. В современных вопросно-ответных системах данный модуль представляет собой классическую поисковую машину, на вход которой поступает запрос из ключевых слов. После того как получен набор текстовых документов, релевантных запросу, извлекаются фрагменты, в которых велика вероятность получения ответа на вопрос. Для того чтобы получить фрагменты из документов, которые могут содержать ответ с наибольшей вероятностью, текст документа делится на части – одним из способов является деление на абзацы. Затем выбирается тот фрагмент (абзац), который содержит все ключевые слова или наибольшее их количество. В системе вопросно-ответного поиска, разработанной в Южном Методистском Университете США [23] применяется немного другой способ получения фрагментов (параграфов) документов для последующего извлечения из них ответа, который рассматривает чуть более сложный случай выбора отрывка текста. Пусть имеется поисковый запрос, состоящий из следующего набора ключевых слов : {k1,k2,k3,k4}. Текст документа разделен на фрагменты (параграфы) и один из параграфов содержит включения k1,k2,k3, причем k1 и k2 встречаются два раза, k3 – один . Вводится понятие окна параграфа – оно включает в себя весь текст между двумя ключевыми словами – одним, расположенным выше остальных по тексту, вторым – ниже. Рассматриваются всевозможные включения ключевых слов во фрагмент документа (окно параграфа). Таким образом, можно для данного случая можно получить 4 случая окна параграфа:

[k1-1, k2-1, k3], [k1-2, k2-1, k3], [k1-1, k2-2, k3], [k1-2, k2-2, k3].

Каждое из окон параграфов оценивается - для каждого из них рассчитываются следующие величины:

1.Same_word_sequence_score – количество слов из вопроса, встречающихся в окне параграфа в таком же порядке;

2.Distance_score – количество слов, разделяющее самые удаленные ключевые слова в окне параграфа;

3.Missing_score: количество слов из запроса, не встречающихся в окне параграфа.

Далее происходит сортировка и выбор фрагмента документа, причем сравниваются величины всех окон всех параграфов.

В некоторых вопросно-ответных системах используется булевский тип поиска информации [6] . Поисковый запрос состоит из слов и булевых операция AND, OR и NOT, в зависимости от которых производится поиск тех или иных слов в документах. Данный подход поиска использовался в системах [6][21]. Система Falcon[22] показала хорошие результаты на конференции TREC 2001 используя поисковую машину SMART булевского типа . Преимущество булевского поиска в том, что можно хорошо настроить процесс поиска формируемым запросом. Но если результаты поиска оказались не совсем удачными и не нашлось ответа, то запрос формируется заново с модификациями. Это называется циклическим поиском, такой приём использовался в системе Falcon.

Некоторые системы вместо классического поиска по ключевым словам используют на этом этапе поиск по индексированной коллекции. Текстовая коллекция предварительно аннотируется такими метками, как классы именованных сущностей и классы синтаксических связей индексируется по ним, что существенно ускоряет поиск.

Во многих современных исследовательских вопросно-ответных системах данный модуль представляет собой элемент архитектуры,

использующий возможности различных поисковых систем, в том числе и популярных веб-поисковых (Yahoo, Bing, Google и т.д.). Так в системе Diogene используется поисковую машину ZPrise [2] [11], в системе Lasso[23] изначально для поиска использовалась Lucene [10]. Задача информационного поиска для вопросно-ответных систем состоит в получении при запросе списка релевантных ссылок и фрагментов текстовых документов - сниппетов. Как правило, сниппеты содержат контекст, в котором встретились ключевые слова в тексте на странице. Просмотрев сниппет, можно приблизительно понять, соответствует ли страница именно вашему запросу, даже не открывая самой этой страницы. Как уже говорилось, это решение связано с тем, что не требуется заново разрабатывать решения информационного поиска.

Многие вопросно-ответные системы для английского используют словари-тезаурусы для расширения множества своих запросов[3][6][15][11][24]. Известной мощной системой для анализа слов является WordNet[7]. WordNet - это мощный толковый словарь и тезаурус, выдающий результаты своей работы (справку по данному слову) в удобном для компьютерного анализа структурированном виде. Сами словарные статьи при этом представляют собой тексты на английском языке, предназначенные для чтения человеком. В настоящее время идет перевод данного словаря на русский язык с последующей его адаптацией для использования при решении прикладных задач.

2.3.3 Извлечение ответа

На данном шаге распознается и извлекается из полученных текстовых фрагментов ответ на вопрос. Важную роль в выделении ответа играет тип ответа. Общая идея решения задачи извлечения ответов состоит в

[7] http://wordnet.princeton.edu/ , WordNet — это электронный тезаурус/семантическая сеть для английского языка, разработанный в Принстонском университете и выпущенный вместе с сопутствующим программным обеспечением под свободной лицензией.

следующем: выявляются так называемые кандидаты для ответа (англ. «answer candidate») [3][11][15] – слова или словосочетания, которые могут рассматриваться как ответ на вопрос; затем производится анализ списка кандидатов, все кандидаты оцениваются и выбирается самый подходящий, то есть тот, который имеет наивысшую оценку.

2.3.3.1 Выбор кандидатов ответа

Выделение именованных сущностей

Стратегия извлечения ответа из текстового фрагмента зависит от типа ожидаемого ответа. Зачастую для фактографических вопросов при первичном анализе фрагментов текста используются методы выделения именованных сущностей [10]. Например, для таких типов ответа, как географические местоположения, имена людей (PERSON, LOCATION, COUNTRY) в извлечении ответа будут использованы алгоритмы распознавания имён собственных [5]. Для извлечения сущностей может использоваться готовая сторонняя система извлечения информации, которая обучается размеченным текстовым корпусом, также может использоваться словари, содержащие списки различных сущностей [11].

Использование шаблонов

Рассмотрим один из популярных способов извлечения ответа – с помощью соответствия шаблонам (англ. «pattern matching») [20]. Для каждого типа ответа составляются шаблоны, с помощью них в текстовых фрагментах производится поиск и выделение кандидата ответа. Для выбора ответа используются информация о типе ожидаемого ответа, полученная на первом этапе работы системы, и символьные шаблоны.

Шаблоны можно создавать как вручную, так и автоматическими обучаемыми алгоритмами[2][15]. Также можно использовать автоматические методы для выявления шаблонов для последующего их применения. Целью

обучения является выявления и построение связей между конкретным типом ответа (например, DATE_OF_DEATH) и конкретным фокусом вопроса (для этого случая - персона). Таким образом, нужно выявить шаблоны, связывающие два вида этих фраз (PERSON/DATE_OF_DEATH). Приведем примерный алгоритм обучения для выявления шаблонов [20]:

1.для создания связи между двумя сущностями создаётся список из правильных пар;

2.производится запрос поисковой машине из частей этих пар;

3.далее выбираются предложения из релевантных документов, содержащих обе части пар;

4.извлекается шаблон, содержащий слова и знаки пунктуации между этими частями пар;

5.далее оцениваются и выбираются шаблоны.

Рассмотрим идею оценки шаблонов. Например, оценка и выбор шаблонов можно произвести следующим образом: составляется запрос поисковой машине из фраз, входящий в вопрос и подходящих по паре (PERSON); далее к найденным фрагментам документов применяется шаблон, и так как правильное значение ответа уже известно, то просто выбираются такие шаблоны, которые имеют высокий процент правильно найденных ответов.

Далее представлены примеры шаблонов, найденные с помощью этого алгоритма.

<NAME> (<DATE_OF_BIRTH> – <DATE_OF_DEATH>)

<NAME> was died on <DATE_OF_DEATH>

Использование N-грамм.

Другим способом извлечения ответов из фрагментов является выявление кандидатов применением n-грамм [6]. N-грамма – это подпоследовательность из n элементов, следующих друг за другом в данной

последовательности. Данный алгоритм эффективно применять к сниппетам при поисковом запросе, полученном при перефразировании вопросительного предложения [4] [6]. На первом этапе из сниппета извлекаются униграммы, биграммы и триграммы. Далее им присваиваются веса, равные количеству сниппетов, в которых встретилась данная n-грамма. Следующий этап – оценивание и сбор кандидатов из n-грамм. При оценивании преследуется цель определения того, насколько данная n-грамма соответствует типу ожидаемого ответа. Далее n-граммы ранжируются, выбирается определенное их количество с высокими оценками и строится кандидат ответа, путем конкатенации n-грамм. Кандидат для ответа с высокой оценкой выбирается в качестве ответа.

2.3.3.2 Оценка кандидатов ответа

После выбора кандидатов производится оценка и выбор потенциального ответа. Оценка производится с помощью проверки (англ. «answer candidate proofing») различными способами[1][4][5]. Для каждого типа ответа составляются шаблоны, с помощью них в текстовых фрагментах производится поиск и выделение кандидата ответа. Есть много различных подходов, применяемых для решения задачи выбора ответа из кандидатов, в целом они все отличаются тем, как формализуются, сравниваются и обрабатываются в них предложения на естественном языке для оценки и выделения потенциальных ответов среди кандидатов ответа.

Метод мешка слов

Один простых способов вычисления оценки для кандидата ответа является метрика метода мешка слов [8]:

$$S_k = \frac{|Q_k \bigcap A_k|}{|Q_k|} \quad (1),$$

где Q_k – множество слов вопроса, A_k – множество слов сниппета, содержащего кандидата ответа, а оценка S_k для k-го кандидата вычисляется, как отношение мощности пересечения множеств Q_k и S_k и мощности множества Q_k. В работе [19] использовали похожую, но более жестко отсеивающую оценку метода мешка слов (q_i – i-ое слово в вопросе, a_j – j-ое слово в сниппете):

$$S = \sum_{i<|Q|, j<|A|}^{|Q|} \frac{match(q_i, a_j)}{|Q|}, \quad (2)$$

$match(q_i, a_j) = 1$,если $q_i = a_j$, 0- иначе.

Метод с использованием грамматик зависимостей предложений.

Другим способом оценки кандидата ответа является метод, использующий грамматики зависимостей предложений[8] (англ. «dependency grammars»). На рисунке 7 представлен пример такого дерева. В работе [18] используется оценка с помощью сравнения множества грамматик зависимостей в виде деревьев. Для вопросительного предложения и предложения в сниппете, содержащего кандидата ответа, строятся так называемые деревья грамматик зависимостей, отражающие связи между словами, рассматривая их как частями речи. Далее они сравниваются, и вычисляется для каждого кандидата оценка неточного сравнения деревьев и выбираются кандидаты с наименьшей оценкой:

[8] Грамма́тика зави́симостей — одна из формальных моделей, разработанных в рамках структурного синтаксиса(наряду с грамматикой составляющих). Представляет строй предложения в виде иерархии компонентов, между которыми установлено отношение зависимости. Таким образом, структура предложения рассматривается в терминах вершин и зависимостей.

$$S = argmin_{a_i \in A} DR(q, a_i), \quad (3)$$

A – множество предложений с кандидатами ответа, q – дерево вопросительного предложения, a_i – предложение с кандидатом ответа, DR – функция оценки неточного сравнения деревьев.

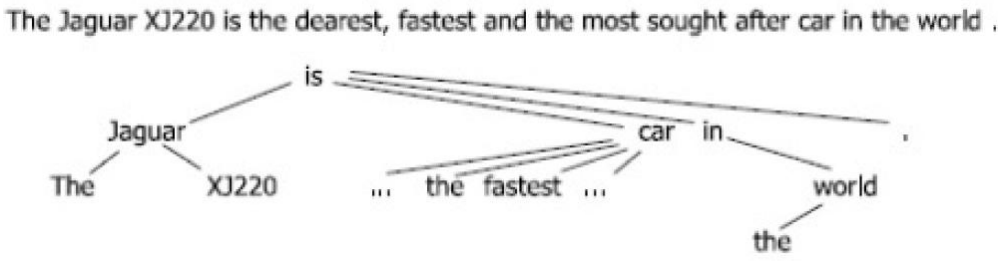

Рисунок 7. Пример дерева грамматических зависимостей для предложения «The Jaguar XJ220 is the fastest car in the world.» [18]

Для неточного сравнения деревьев зависимостей используется метод оценки сложности преобразования одного в другое. Введены три операции для преобразования дерева: удаление вершины, вставка вершины, изменение вершины. Каждая такая операция s_i имеет некоторую стоимость $\gamma(s_i)$. Задача состоит в нахождении последовательности операций $S = <s_1, s_2, \ldots, s_n>$, преобразующее дерево предложения T_1 из сниппета в дерево вопросительного предложения T_2 и имеющее наименьшую суммарную стоимость:

$$\gamma(S) = \sum_{i=1}^{n} \gamma(s_i), \quad (4)$$

Таким образом, для двух деревьев находится оценка минимального редактирования:

$$\delta(T_1, T_2) = min(\gamma(S)|S(T_1) = T_2), \quad (5)$$

где минимум ищется по всем S.

Авторы провели оценку метода на 454 пар вопросов и ответов и сравнивают свой метод с методом оценки с помощью модели мешка слов, результаты приведены в таблице 2 в виде процентного отношения корректно обработанных пар к общему числу тестовых данных.

Таблица 2. Сравнительные результаты тестов методов извлечения ответов.

Метод	Корректно обработанных вопросов, %
Метод неточного сравнения деревьев грамматик зависимостей	40.31
Метод мешка слов	33.26

Метод с использованием семантических структур

Для оценки кандидата ответа можно использовать обработку текстовой информации, связанной с разметкой семантических ролей [12] (англ. «semantic role labeling»). Общая формулировка задачи разметки семантических ролей состоит в следующем: для предложения на естественном языке необходимо определить множество участников ситуации, описываемой в этом предложении, и их семантические роли – соответствующие отношения между участниками. В результате создается семантическая структура – структурированное представление текстовой информации, представляемое в виде ориентированного графа, которое используется для оценки кандидата ответа. Для этого сравниваются граф вопросительного предложения и граф предложения, в котором содержится

кандидат ответа. И по вычисленной мере схожести графов можно проставить оценку кандидату ответа.

Для русского языка из доступных решений, которые можно использовать для оценки кандидата ответа на вопрос, есть технология семантического анализа системы АОТ[17]. Семантический анализ предложения подразумевает процесс выявления его семантической структуры. Введем терминологию, определенную авторами системы АОТ и аналогичную используемой в задаче разметки семантической ролей. Семантическая структура состоит из семантических узлов (участников ситуации) и семантических отношений (семантических ролей). После анализа предложения получается множество из узлов и отношений между ними. Узлами являются слова или словосочетания из предложения, а отношения представляют собой связи с метками, обозначающими различные типы семантических ролей. В результате полученное множество можно визуализировать в виде ориентированного дерева-графа, который авторы системы называют семантическим графом предложения. На рисунке 8 представлен пример разбора предложения в виде семантического графа.

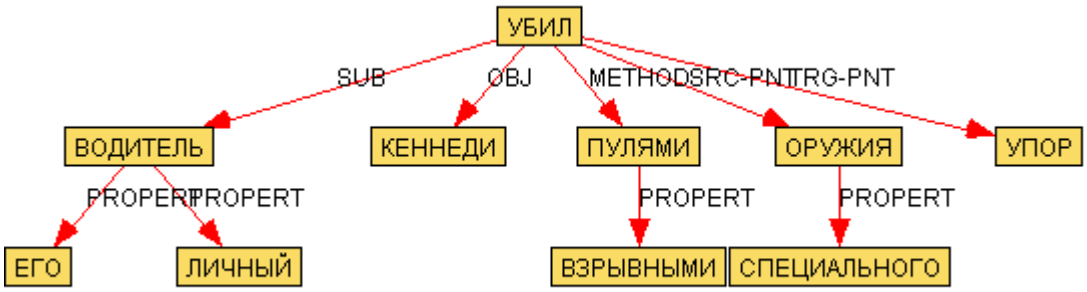

Рисунок 8. Семантический граф для предложения

«Кеннеди убил его личный водитель из специального оружия взрывными пулями в упор.»

2.4 Выводы

Большинство исследовательских вопросно-ответных систем построено по типовому конвейеру. Различают этап анализа вопроса, на котором важно определение типа ожидаемого ответа, этап информационного поиска, на котором получают текстовые фрагменты, и этап извлечения ответа, на котором производится поиск и выбор кандидатов ответа , их оценка и выбор потенциального ответа. Бóльшая часть современных вопросно-ответных систем используются различные методы анализа и обработки естественных языков. На основе исследования существующих решений можно сделать вывод о возможности реализации прототипа вопросно-ответной системы для русского языка, основанной на веб-поиске с использованием методов обработки естественного языка, так как при этом нет необходимости в поддержке большой базы текстовых документов, отсутствует необходимость в привлечении экспертов, не нужна разработка методов поиска текстовых фрагментов, содержащих потенциальный ответ и можно абстрагироваться от задач информационного поиска.

3 Исследование и построение решения задачи

При разработке вопросно-ответной системы за основу была взята типовая архитектура вопросно-ответной системы, основанной на использовании веб-поиска (англ. «web-based»). В главе 2.2 описывается общая архитектура таких систем, их преимущества и недостатки. Кроме перечисленных достоинств так же можно отметить следующие:

- отсутствие необходимости поиска и выделения текстовых фрагментов;

- нет необходимости создания, хранения и разработки способа индексирования большого количества текстовой информации;

- высокая архитектурная гибкость системы;

- высокая скорость реализации;

- приемлемость для экспериментальных исследований.

Недостатки данной архитектуры систем несущественны для поставленной в работе задачи.

Задачу разработки архитектуры и создания прототипа вопросно-ответной системы можно разделить на следующие подзадачи, при котором производится поиск ответа на вопрос:

1) подзадача анализа вопроса;
2) подзадача информационного поиска;
3) подзадача выделения и обработки потенциальных ответов.

Первая подзадача подразумевает ввод вопроса на естественном языке и обработки и формализация предложения различными анализаторами (синтаксическим, морфологическим, семантическим), определяются соответствующие его атрибуты для дальнейшего их использования. На этапе информационного поиска необходимо произвести поиск и анализ документов - отбираются документы и их фрагменты, в которых может содержаться ответ на исходный вопрос. Третья подзадача предполагает извлечение ответа: система, получая текстовые документы или их фрагменты, извлекает из них

слова, предложения или отрывки текста, которые могут стать ответом.

3.1 Анализ вопроса

В процессе исследования решений в области вопросно-ответного поиска, было выяснено, что большинство исследовательских систем поддерживают заранее определенные типы вопросов и не могут отвечать на вопросы причины и образа действия – вопросы типа «Почему…?», «Как…?».

В данной работе были рассмотрены фактографические вопросы о персонах и топонимах, общие вопросы о месте. В системе введена таксономия типов ожидаемого ответа, показанная на рисунке 9, по аналогии с примером системы [11]. На нем изображена иерархия типов согласно соответствующим типам шаблонам и дальнейшей стратегии поиска ответа на вопрос.

В таблице 3 представлены используемые типы ожидаемого ответа в разработанной системе. Такое деление ответа на типы связано с методами поиска и выделения ответа, которые будут рассмотрены в разделе 4.3.

Изначально после ввода пользовательского запроса предложение проверяется на корректность. Рассматриваются только правильно введенные со следующей точки зрения вопросы:

- предложение не содержит знаков препинания, кроме запятой;
- предложение написано на русском языке;
- предложение вопросительного типа;
- предложение соответствует одному из рассматриваемых в работе шаблонов.

Вопрос считается некорректным, если не удовлетворяет выше перечисленным условиям. Некорректные вопросы не рассматриваются

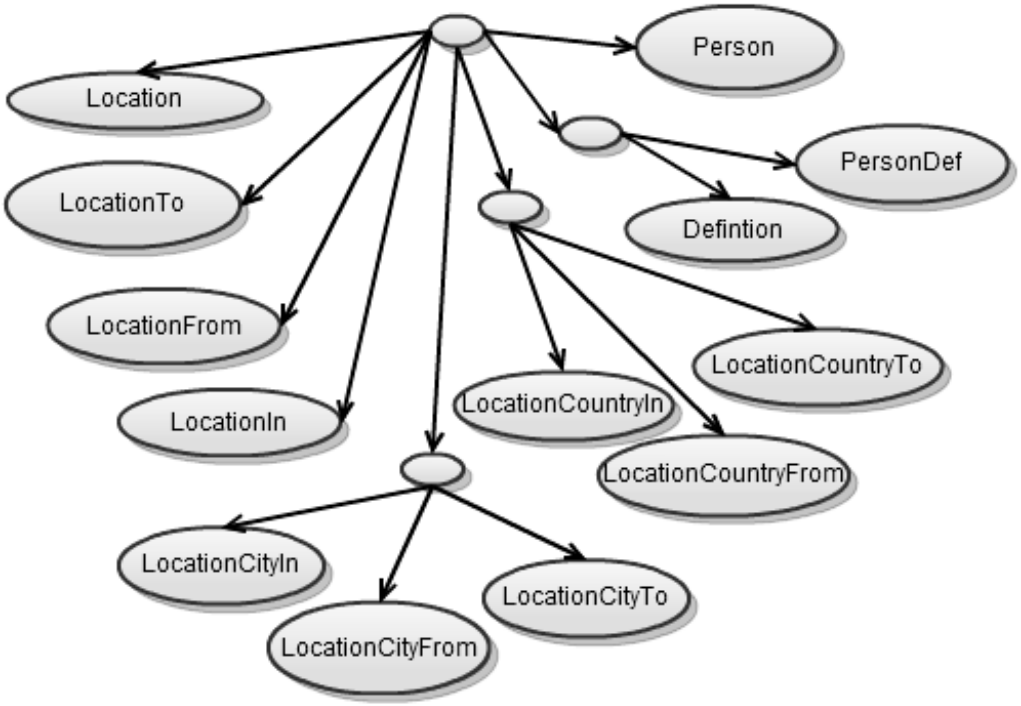

Рисунок 9. Таксономическое дерево типов ожидаемого ответа

Таблица 3. Типы ожидаемого ответа и соответствующие регулярные выражения

Тип ожидаемого ответа		Регулярное выражение для шаблона
LocationIn		(Где находится | Где располагается | Где было) [^?]*?
LocationTo		Куда [^?]*?
LocationFrom		Откуда [^?]*?
LocationCountry	LocationInCountry	В какой стране [^?]*?
	LocationFromCountry	Из какой страны [^?]*?
	LocationToCountry	В какую страну [^?]*?

LocationCity	LocationInCity	В каком городе [^?]*?
	LocationToCity	В какой город [^?]*?
	LocationFromCity	Из какого города [^?]*?
Location		Где [^?]*?
Definition		(Что такое | Чем является | Что есть) [^?]*?
PersonDef		(Кто такой | Кем был | Кем является) [^?]*?
Person		Кто [^?]*?

Далее необходимо сформировать поисковые запросы для подзадачи информационного поиска. В работе [6] авторы используют части предложения без вопросительных слов. Так же поисковые запросы формируются при помощи перефразирования вопросительного предложения в утвердительное . В данной работе были использованы перечисленные выше методы формирования запросов, но как показали тесты, формирование запроса в виде полностью самого вопросительного предложения показывают более релевантные результаты и большую вероятность и точность нахождения правильного ответа. Это связано с тем, что современные алгоритмы информационного поиска ориентируются на пользовательские запросы в виде предложений.

В данной работе был рассмотрен метод расширения множества запросов с помощью синонимов. Чтобы не менять смысл вопроса необходимо использовать синонимы глаголов, а не существительных и прилагательных. Для этого необходимо провести синтаксический разбор

предложения, выделить сказуемое в форме глагола. Далее используя начальную форму глагола, производится поиск его синонима (-ов). Для синтаксического разбора можно использовать один из обучаемых парсеров с использованием размеченных языковых корпусов. Для русского языка существует на данный момент только один синтаксически размеченный корпус - СинТагРус и поддерживающий его парсер MaltParser[9]. Для выделения частей речи были рассмотрены результаты морфологического парсера системы Диалинг, которая базируется на грамматическом словаре А.А.Зализняка. В процессе морфологического разбора для каждого слова входного текста представляется множество морфологических интерпретаций следующего вида: начальная форма слова, морфологическая часть речи.

После того, как каждое слово было морфологически разобрано, выделяются глаголы. Далее используется инфинитив – начальная форма глагола – для поиска синонимов. Обычно для получения синонимов используются различные словари. В процессе разработки были попытки использовать частично готовые элементы словаря WordNet для русского языка, но словарь не полностью готов и не адаптирован для эффективной работы. Так, например, можно было расширить множество поисковых запросов применением синсетов – синонимический рядов.

Так же на этапе анализа вопроса производится семантический анализ предложения – подробнее об этом будет рассказано в разделе 3.3.

3.2 Информационный поиск.

Для этапа информационного поиска были рассмотрены несколько самых популярных современных поисковых систем, а именно следующие: Yandex[10], Google[11], Bing[12], Yahoo[13]. Каждая поисковая система имеет свои

[9] http://www.maltparser.org/

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

особенности, и качество полученного результата зависит от предмета поиска и точности формулировки запроса. Поэтому, приступая к поиску информации, прежде всего, нужно четко представлять себе, что именно нужно найти. Современные системы информационного поиска, изначально ориентированные на английский язык, системы поражают числом проиндексированных документов. Однако для поиска информации на русском языке, особенно в российской части Интернета, лучше приспособлены русскоязычные поисковые машины.

Системы Yahoo и Bing выдают по сравнению с ними выдают менее релевантные результаты. Для русского языка Bing возвращает сниппеты очень малого объема – обычно одну строку. Причем сниппеты данной системы могут состоять из ненужных в данном случае различных тегов страниц – перечислений существительных. В ходе исследования и разработки было выяснено, что это встречается в 4-5 сниппетах из первых 20 результато. В системе Yahoo при русскоязычном запросе сниппеты возвращаются также очень маленькой величины для нашей задачи – чаще всего в 1 строку.

Так как рассматривается задача вопросно-ответного поиска для русского языка, то были выбраны наиболее адаптированные для данного языка системы, имеющие возможность предоставления результатов поиска: Google и Yandex. Во-первых, они специально ориентированы именно на русскоязычные ресурсы Интернет и, как правило, отличаются большей полнотой охвата и глубиной исследования этих ресурсов. Во-вторых, российские системы работают с учетом морфологии русского языка, то есть в поиск включаются все формы искомых слов. Данные системы лучше

[11] http://www.google.ru/

[12] http://www.bing.ru/

[13] http://ru.yahoo.com/

учитывают и такую исторически сложившуюся особенность российских Интернет-ресурсов, как сосуществование нескольких кодировок кириллицы. Это стало причиной выбора в пользу двух других систем. Наиболее объемные сниппеты выдаются системой Yandex, поэтому предпочтение в некоторых случаях будет дано сниппетам этой системы.

На данном этапе изначально выбирается определенное количество первых сниппетов из результатов поиска данных систем. Формируется массив сниппетов с соответствующими ссылками на веб-страницы, позициями в результатах поиска, и массив передаётся на следующий этап – этап поиска и извлечения ответа.

3.3 Поиск и извлечение ответа

На данном этапе известен тип ожидаемого ответа и имеется некоторое количество сниппетов. Задача состоит в поиске и выделении кандидатов ответа из сниппетов, их оценке и выборе нескольких кандидатов с максимальной оценкой в качестве потенциального ответа на вопрос.

Изначально сниппеты оцениваются по принципу модели «мешка слов» в сравнении с предложением, введённым пользователем. Для каждого сниппета вычисляется следующая оценка:

$$BOW(Q,S) = \frac{|Q \cap S|}{|Q|}, \quad (6)$$

Далее сниппеты упорядочиваются согласно оценке и позиции релевантности сниппета в результатах, выданных системами поиска. То есть если сниппеты имеют равную оценку, то ранжируются они по позиции.

3.3.1 Формирование списка кандидатов ответа

Далее над каждым сниппетом извлекаются все именованные сущности – слова или словосочетания, определяющие персоны, организации, географические объекты и прочие объекты, обозначаемые в тексте с использованием имен собственных. После этого выбираются только те именованные сущности, которые соответствуют типу ожидаемого ответа, и добавляются в список кандидатов ответа. Список кандидатов ответа состоит из пар вида <<кандидат ответа>,<сниппет>>, содержащий извлеченную сущность и сниппет, соответствующий ему. Далее представлен список типов ожидаемого ответа, для которых используется извлечение именованных сущностей на этапе формирования списка кандидатов ответа:

- <*PersonCommon*> - имена (полные) людей;
- <*LocationCountry*> - географические топонимы, обозначающие названия стран;
- <*LocationCity*> - географические топонимы, обозначающие названия городов;
- <*LocationCommon*> - все географические топонимы, в их число входят названия стран, городов, континентов, организаций.

Для выделения именованных сущностей можно использовать возможности различных технологий обработки естественного языка для распознавания именованных сущностей (англ. «named entity recognition»)

После того как сформирован список кандидатов ответа, необходимо его оценить и выбрать из него потенциальные ответы.

3.3.2 Семантический анализ предложений

Как уже было указано в разделе 2.3.3 существует несколько подходов к оценке кандидатов ответа. В данной работе рассматривается метод использования информации о семантической структуре предложений и анализ отношений между его узлами (см. раздел 2.3.3). Для этого использовались возможности технологии семантического анализа системы АОТ.

Адаптация семантических структур

В соответствующих семантическим структурам графах могу быть так называемые абстрактные узлы (узлы типа «mua», «mna», «copul», «modulcopul») , введенные авторами системами, которые в данной работе не имеет смысла использовать, так как такие узлы не содержат слова и нужную информацию, и играют роль в указании перечисления, замены и др. Поэтому графы после получения подвергаются обработке. В графах удаляются данные узлы, которые не содержат слов и словосочетаний, а входящие и исходящие связи заменяются на такие же, но без прохода по этим абстрактным узлам, то есть связывают соседние им узлы. На рисунке 10 приведены примеры графов для предложений, в данных примерах содержатся упомянутые абстрактные узлы. Далее на рисунке приведен пример изменения структур в виде графов.

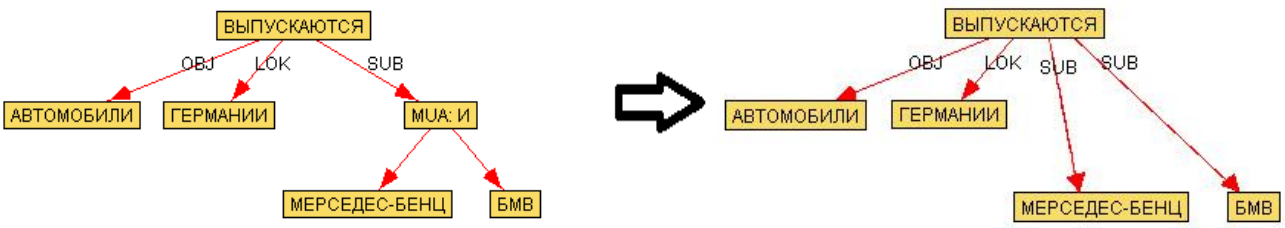

Рисунок 10. Пример изменения графа для предложения *«Автомобили "БМВ" и "Мерседес-Бенц" выпускаются во Германии.»*.

Также при изменении графов узлы, которые имеют исходящую связь типа «THE SAME», удаляются, а входящие в них связи переводятся на те узлы, к которым идут связи от удалённого. На рисунке 11 показан пример изменения графа в случае наличия такой связи в графе. Данное изменение, значительно улучшает поиск и оценку кандидатов ответа с типом «PersonCommon».

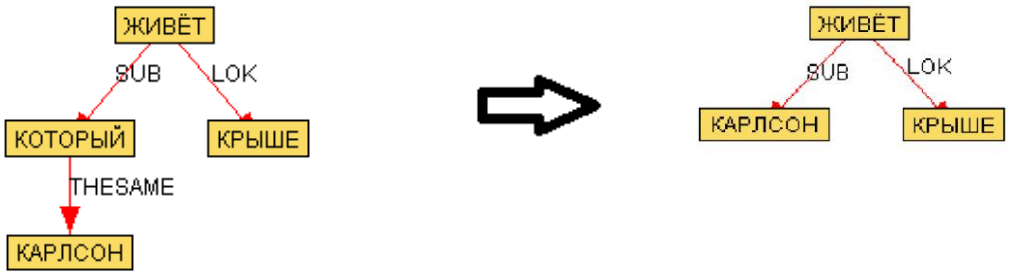

Рисунок 11. Семантический граф со связью типа «THE SAME» для предложения *«Карлсон, который живёт на крыше.»*.

Сравнение графов.

Перед тем как сравнивать семантические графы предложений с целью оценки кандидата ответа, необходимо выделить предложение из сниппета, так как сниппет изначально может содержать несколько различных предложений и частей текста. Для этого было разработано регулярное выражение, с помощью которого выделялись предложения из сниппета:

$$(\text{\^{}}|(?<=[.!?]\\s))(\\d+\\.\\s?)*[А\text{-}ЯA\text{-}Z][\text{\^{}}!?]*?[.!?](?=\\s*(\\d+\\.\\s)*[А\text{-}ЯA\text{-}Z]|\$)\,, \quad (9)$$

Далее выбирается предложение, в котором может содержаться ответ. Данный шаг делается тривиальным образом: выбирается предложение, содержащее большее количество ключевых слов поискового запроса. После

того, как извлечено предложение, для него строится семантический граф. Данный шаг происходит с вычислением меры по принципу мешка слов по формуле, аналогичной (7). Выбирается предложение с максимальной мерой.

В работе [16] авторы предлагают метод параллельного обхода и сравнения графов предложений, в качестве решения задачи валидации ответов. В ходе исследований был использован алгоритм, основанный на таком же решении, с различными модификациями, представлен далее.

Алгоритм сравнения графов:

1) в графе вопросительного предложения ищется узел, соответствующий вопросительному слову («КТО», «ГДЕ»);
2) в графе предложения сниппета ищется узел, соответствующий кандидату ответа;
3) выполняя операции, аналогичные поиску в глубину, продвигаемся одновременно по обоим графам от исходных узлов по таким связям и узлам, что тип связи из графа вопросительного предложения должен совпадать с типом связи из графа предложения сниппета;
4) при совпадении узла и/или связи производится инкремент оценки:
   a) случай совпадение типов связей: инкремент на 1;
   b) случай совпадения слов, соответствующих узлам: инкремент на1;
      под совпадением узлов понимается случаи равенства строк в узлах и случай при котором, строка в одном узле является подстрокой в другом;
5) накопленная оценка умножается на вес k (значение которого можно изменять) и полученное значение прибавляется к общей оценке кандидата ответа[14].

В ходе тестирования метода было установлено, что данный предложенный метод показывает невысокую полноту при малом количестве

[14] О вычислении общей оценки кандидата ответа подробнее будет рассказано далее.

сниппетов. При количестве сниппетов равном 10 в лучшем случае выделяются и подвергаются сравнению 1-2 предложения из сниппетов. Так же данный метод увеличивает время работы, так как вычислительно сложен и малоэффективен.

Причина заключается в том, что в сниппетах ответы могут содержаться не в предложениях, обозначенных в явном виде, системы информационного поиска включают в выдаваемые сниппеты фрагменты предложений. В сниппетах часто встречаются фрагменты из разных мест текстового документа, при этом в сниппетах не соблюдаются орфографические и пунктуационные правила, пунктуационные знаки либо могут отсутствовать, либо могут стоять друг за другом и не иметь смысла. Например, далее представлены сниппеты, содержащие точный ответ на вопрос «Где живут пингвины?», но не содержащие предложения в явном виде.

«Пингвины. … - но многие виды пингвинов живут в относительно теплых местах, где даже снега не бывает, — на юге Австралии, в Новой Зеландии, на юге Африки и на»

«. – пингвины… где обитают эти необыкновенные птицы? «конечно же, пингвины живут в Антарктиде!...» – ответите вы…»

Так же был исследован способ без выявления предложений из сниппета. На выходе после семантического анализа получаются несколько графов, а в некоторых случаях и просто одиночные несвязанные узлы, вследствие чего не получается применить алгоритм сравнения графов. Поэтому от данного метода было принято решение отказаться в виду требования производительности, большого времени отклика, сложности и малой эффективности алгоритма при невысокой полноте результатов.

Использование семантической структуры сниппетов.

В ходе исследования был разработан оригинальный метод оценки кандидата ответа с использованием семантических связей между словами. Данный метод заключается в поиске кандидата ответа в сниппете и вычислении оценки L с помощью информации о семантическом отношении между словами. Алгоритм оценки состоит в следующем:

1) L=0;
2) сниппет подвергается семантическому анализу;
3) в семантической структуре сниппета ищется узел U, соответствующий кандидату ответа, то есть содержащий из него слова или словосочетания;
4) затем производится выявление входящих в U связей;
5) если существует входящая связь, соответствующая (по таблице 4) типу ожидаемого ответа в данном случае, то к оценке L прибавляется 0.5;
6) если есть узел, связанный с U и являющийся глаголом-сказуемым близким по смыслу к глаголу-сказуемому из вопросительного предложения, то к оценке L прибавляется 1; данный шаг может быть проделан с помощью вычисления метрики по словарю WordNet; в данной работе к L прибавляется 1, если инфинитивы глаголов совпадают.

Затем оценка L умножается на весовой коэффициент k. В данной работе значении k было равным 10, что позволяет выбрать в качестве ответа кандидата, удовлетворяющего условиям 5) и 6).

Таблица 4. Типы семантических связей, соответствующих типам ожидаемого ответа

Тип ожидаемого ответа	Тип семантической связи
Location, LocationInCountry, LocationInCity	LOK
LocationFromCity, LocationFrom, LocationFromCountry	SRC-PNT
LocationToCity, LocationTo, LocationToCountry	TRG-PNT
Person	AGENT, SUB, F-ACT

Поиск ответа по семантическим дуплетам

В ходе исследования и разработки было выяснено, что некоторые вопросы, могут подразумевать не только поиск именованных сущностей. Например, на вопрос *«Где живет Карлсон?»*, ответом может быть словосочетание *«на крыше»*, а не только слова, обозначающие город или страну *(«Стокгольм», «Швеция»*). Или же вопрос, не подразумевающий ответа, являющегося именованной сущностью: *«Где находится самая маленькая кость человека?»* - ответом будет словосочетание *«в среднем ухе»*. Такие случаи распознаются «запасной стратегией» поиска ответа по семантическим дуплетам и рассматриваются в системе в том случае, если количество кандидатов ответа меньше какого-либо числа – заранее определенного порога K – то дальнейшая стратегия поиска ответа состоит в извлечении ответов из семантических структур предложений сниппета. Порог определяется экспериментальным способом, чаще всего он может

(7)

быть пропорционально связан с числом рассматриваемых сниппетов. Например, в разработанной системе это число S равно:

$$S = \frac{N}{n},$$

где n – настраиваемый параметр.

Алгоритм поиска ответа заключается в следующем:

1) в семантической структуре сниппета выделяются дуплеты следующего вида:
   <сказуемое-глагол> <зависимая часть>, где тип связь между ними соответствует типу ожидаемого ответа (таблица 4);
2) зависимая часть целиком (в общем случае она является поддеревом в семантическом графе) выделяется в список кандидатов ответа;

Вопросы об определениях.

В данной работе рассматриваются вопросы, требующие выдачи определения, как уже было написано, поддерживаются вопросы, соответствующие регулярным выражениям в таблице 5 для типов ожидаемого ответа «PersonDef» и «Definition». В вопросе выделяется так называемый концепт – та часть вопроса, следующая за вопросительными словами в шаблонах распознавания типа ожидаемого ответа. Например, в вопросе *«Кто такой Жорес Алфёров?»* концептом будет *«Жорес Алфёров»*. В список кандидатов ответа включаются описания - слова или словосочетания, найденные в сниппете с помощью шаблонов, описанных в таблице 5, то есть соответствующие метке «описание» в них.

Таблица 5. Шаблоны для поиска кандидатов ответа с соответствующими типами ожидаемого ответа.

Тип ожидаемого ответа	Шаблон поиска
PersonDef, Definition	<концепт> - <описание> <концепт>,<описание>, <описание>, такой(ая, ое) как <описание>

В работе системы при извлечении ответа применяется следующий эвристический приём, позволяющий находить ответы на описанные вопросы. На этапе получения ответа на вопросы такого рода, если в первых 10 результатах есть ссылки на статью в Википедии, то ответом станет первое предложение в абзаце статьи по ссылке сниппета. Первый абзац энциклопедической статьи содержит в начале емкое и краткое определение. В вопросах о персоне, в первом абзаце кратко представлена информация о датах рождения и смерти, о роде деятельности и заслугах человека.

Если найдено такое определение, то алгоритм заканчивает работу, оно не оценивается и не включается в список кандидатов ответа, а сразу выдается в качестве потенциального ответа пользователю.

3.4 Оценка списка кандидатов ответа и выбор кандидатов.

После формирования списка кандидатов ответа необходимо оценить и отобрать наиболее достоверные пары (кандидат, сниппет). Для этого к каждой паре (C,S) (C-кандидат ответа, S – сниппет, содержащий его) из списка применяется функция оценки пары, которая учитывает так же изначальный анализ сниппетов. Данная оценка формируется из нескольких факторов, влияющих на её значение, которые будут описаны ниже.

1) Содержится ли кандидат ответа (или его часть, если он состоит из нескольких слов) в вопросительном предложении. Используется переменная D в формуле (10) вычисления общей оценки: D(C,S) = 0, если содержится, D(C,S) = 1 – иначе. Очевидно, что если заданный

вопрос содержит именованную сущность, то кандидат не должен являться им.

Например, один из примеров: для вопроса *«Кто убил Кеннеди?»* кандидатом не должна являться сущность «Кеннеди» или «Джон Кеннеди».

2) Одинаковы ли сказуемые из вопросительного предложения и семантического триплета (или сниппета). Пусть p_q – сказуемое в вопросе, p_s – сказуемое из триплета (или сниппета). $E(p_s,p_q)$ – метрика сравнения двух сказуемых. В данной работе $E(p_s,p_q) = 1$, если начальные формы глаголов равны (инфинитивы), $E(p_s,p_q) = 0$ – иначе. Но можно было бы использовать метрику близости слов из словаря WordNet.
3) Учитывается количество таких же кандидатов полностью или частично идентичных данному (обозначим его как M). Для этого используется следующая составляющая общей оценки:

$$M(C) = \frac{Z(C)}{N}, \qquad (8)$$

где Z(C,S) – количество таких кандидатов, N – общее количество кандидатов в списке.

4) Используется высчитанная при анализе сниппетов значение функции BOW (Q,S).
5) Учитывается оценка L, вычисление которой описано в разделе 3.3.2.

Общая оценка пары кандидата и сниппета равна:

$$O(C,S,Q) = (k_1 * BOW(S,Q) + k_2 * M(C) + k_3 * E(p_s,p_q) + k_4 * L(C,S)) * D(C,S), \qquad (9)$$

где k_i – весовые коэффициенты, значения которых можно задавать при запуске системы.

Затем весь список кандидатов сортируется по оценке O(C,S,Q), при этом из него удаляются все повторяющиеся кандидаты (которые полностью или частично равны), оставляются только те экземпляры, которые имеют максимальную оценку. После этого выдаются в качестве ответа три кандидата ответа с наибольшими оценками.

3.5 Тестирование и оценка системы

В ходе исследования, разработки и тестирования всего было протестировано около 270 вопросов. Для тестирования и оценки системы были созданы экспертами коллекции пар «вопрос, ответ», в которых содержались вопросительное предложение и корректный ответ.

Модуль тестирования системы производит серию запусков системы на тестовых данных. Тестовые данные были структурированы и помещены в текстовые файлы. При оценке работы системы были вычислены такие оценки эффективности, как точность и полнота. Точность работы - это отношение количества правильных ответов на вопросы на общее количество вопросов тестовых данных. Значение полноты равно отношению количества всех выданных ответов на вопросы на общее количество вопросов.

В таблице 6 приведены результаты тестирования без семантического анализа. Всего было обработано при тестировании 102 вопроса, начинающихся на «Где» (в таблице 7 обозначены как «Группа тестовых данных №2»), и 75 вопросов, начинающихся на «Кто» (в таблице 7 обозначены как «Группа тестовых данных №1»). На первой серии тестовых запусков системы не использовался поиск по семантической структуре в виду его медленной работы, на второй - были обработаны только те вопросы, на которые были выданы неверные ответы, и использовался также

поиск по семантической структуре. Для сравнения была протестирована одна существующая на момент исследований система для русского языка AskNet, по причине технических ограничений была протестирована только часть тестовых данных (70 единиц коллекции «Группа тестовых данных № 1»), система показала точность равную 0.51 и полноту равную 0.81. Результаты работы системы, представленной в данной работе, сравнительно лучше.

Таблица 6. Результаты тестов

Серия тестовых запусков	Группа тестовых данных	Количество правильных ответов	Количество неправильных ответов	Количество необработанных вопросов
1	1	44	21	10
1	2	52	38	12
2	1	51	7	17
2	2	62	21	19

Таблица 7. Тестовые запуски с учетом оценки с использованием семантического анализа

Группа тестовых данных	Точность	Полнота
1	0.57	0.86
2	0.51	0.88

Таблица 8. Тестовые запуски без учета оценки с использованием семантического анализа

Группа тестовых данных	Точность	Полнота
1	0.71	0.77
2	0.60	0.81

В ходе тестирования было выяснено, что вклад составляющей D общей оценки сильно влияет на отбор кандидатов ответа, тем самым могут выдаваться некорректные результаты, чаще всего вовсе не выдаётся ответ. Например, это происходит в следующих вопросах (в скобках указаны корректные ответы из тестовых данных):

«Где находится собор Парижской богоматери?» («Париж»);

«Кто основал буддизм?» («Будда»);

«Кто сочиняет народные песни?» («Народ»).

Также ответы не выводятся на те вопросы, для которых поисковые системы не выводят сниппеты близкие по мере «мешка слов» к вопросительному предложению, содержащие потенциальные ответы. Возможно, было бы лучше использовать поисковые системы булевского типа, про которые написано в разделе 2.2.

4 Описание практической части

4.1 Обоснование выбранного инструментария

В качестве основного языка программирования был выбран объектно-ориентированный язык Java по следующим причинам:

- для Java разработано большое количество библиотек;
- для нашей задачи не требуется высокая скорость работы с большими количествам информации непосредственно из памяти.

На этапе информационного поиска использовались возможности поисковых систем Google и Yandex. Для этого применялась java-библиотека Google Ajax. Результаты поиска выдаются в виде структурного массива в текстовом формате данных JSON. Элемент массива представляет собой каждый пункт результатов, имеет поля ссылки, сниппета и заголовка. Для получения результатов поиска Yandex использовалась java-библиотека Yandex.XML. С помощью неё можно получить результат поиск в виде документа формата XML, который далее обрабатывается с помощью SAX-парсера. В итоге строится структурированный массив с результатами поиска. Результаты поиска обеих систем объединяются в единый общий структурированный список, в каждом элементе которого содержится текст сниппета и вся информация о нём (позиция в результатах поиска, вид поисковой машины, адрес веб-страницы, заголовок).

Для извлечения именованных сущностей используется комплект средств разработки для Java (Java SDK) для доступа к возможностям сервиса обработки естественных языков Alchemy API.

Для получения семантической структуры предложений используется библиотека AOT[17]. Библиотека доступна под лицензией LGPL. Для того чтобы воспользоваться методами библиотеки, необходимо использовать COM-объекты, корректный доступ к которым можно получить под Visual Studio. Для этого написано консольное приложение на языке C#, как .NET –

ориентированном аналоге Java. С помощью вызовов COM-объектов создается семантическая структура, которая затем преобразуется (описание преобразований графов представлено в разделе 3.3.2) и сохраняется в файл. После этого файл считывается и используется в основной системе.

Для удобной пользовательской и исследовательской работы был создан графический интерфейс вопросно-ответной системы. При поиске можно изменять количество используемых сниппетов, выбор поисковых машин, параметры оценки кандидатов ответа и весовые коэффициенты k_i. В Приложении А приведен снимок экрана с примером работы разработанной системы.

4.2 Общая схема работы

На рисунке 12 показана диаграмма, на которой изображена общая схема работы системы. Этапы «Выделение кандидатов ответа», «Семантический анализ» рассмотрены в разделах 4.3.1, 4.3.2 соответственно. Этапы «Обработка списка кандидатов и их оценка», «Выбор кандидатов в качестве ответа» описаны в разделе 4.4. Рассмотрим остальные несколько этапов работы системы.

4.2.1 Распознавание типа ожидаемого ответа

Распознавание производится с помощью сравнения вопросительного предложения с текстовыми шаблонами. Для данного этапа используется в системе несколько файлов. В одном файле хранится список всех поддерживаемых типов ожидаемого ответа, в остальных текстовые шаблоны (каждому типу ожидаемого ответа соответствует один файл). Тем самым обеспечивается гибкость системы, несложно добавить новый тип ожидаемого ответа или удалить существующий.

4.2.2 Информационный поиск. Формирование списка результатов поиска

На данном этапе формируется список результатов поиска на основе выданных поисковыми машинами. В элемент списка входят следующие поля:

-сниппет;

-позиция в результатах поиска;

-заголовок результата поиска;

-название поисковой машины.

Подробнее данный этап описан в разделе 4.2.

4.3 Общая архитектура системы

На рисунке 15 показана диаграмма классов. На диаграмме изображена общая архитектура системы.

Класс «Query» является классом управляющим, работой системы, принимает на вход вопрос, отвечает за его обработку, информационный поиск, формирует список кандидатов.

Класс «Question» содержит информацию о вопросительном предложении, содержит список слов из предложения, распознает по вопросу тип ожидаемого ответа (метод «recognizeType»), его семантической структуре (атрибут «graph»), строит список ключевых слов для поискового запроса (метод «extractKeywordList»).

Классы «SearchYandex» и «SearchGoogle» используют API систем информационного поиска, отправляют запросы поисковым системам Yandex и Google соответственно, принимают результаты, структурируют их, формируют массивы с результатами поиска (объекты класса «SearchResults»).

Класс «AnswerCandidate» содержит информацию о кандидате ответа и его оценке.

На рисунке 16 показана диаграмма классов консольного приложения для извлечения и обработки семантических структур текстовых документов.

Класс «Console» являются управляющим классом консольного приложения и организует его работу, производится считывание текстовых фрагментов из входного файла (по директории «inputFile») и запись информации о структуре в выходной файл (по директории «outputFile»).

Класс «SemGraph» содержит информацию о семантической структуре текста, узлам соответствуют объекты класса «Node», отношениям между узлами соответствуют объекты класса «Link». Класс «Morph» используется для извлечения морфологической информации о словах.

4.4 Характеристики функционирования

4.4.1 Время отклика

Компоненты вычислительной машины, на которой проводились тестовые запуски:

-процессор Intel® Core ™ i5-2410M 2300 MHz;

-оперативная память 4096 Mb.

Среднее время отклика системы с использованием графического интерфейса равно 12.343с без учёта времени семантического анализа (вычислено среднее время работы на 10 тестовых запросах).

Среднее время работы консольного приложения для семантического анализа варьируется в зависимости от количества обрабатываемой информации (величины сниппета). Среднее время обработки 20 сниппетов равно 342.320с (вычислено среднее время консольного приложения работы на 10 тестовых запусках).

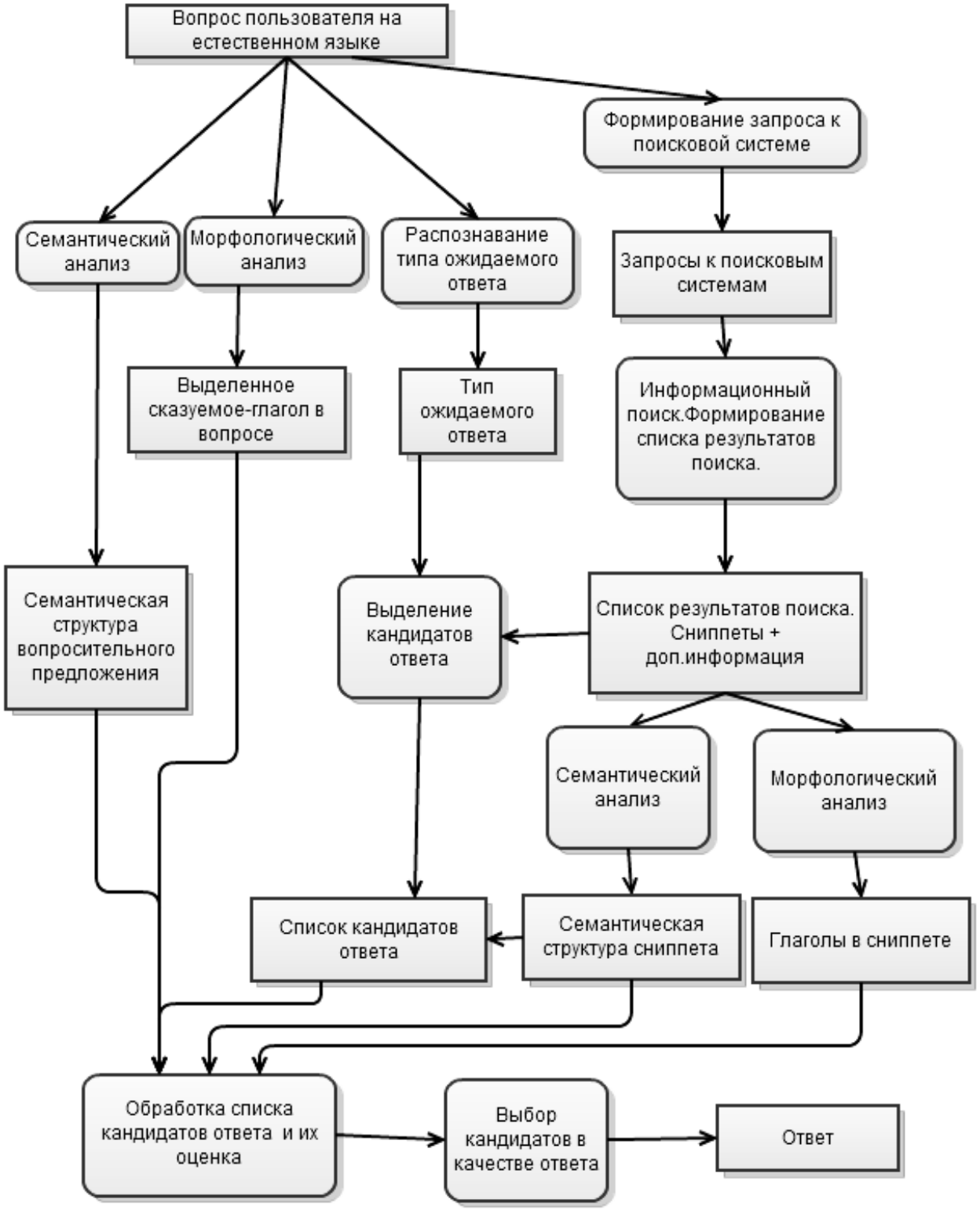

Рисунок 12. Общая схема работы

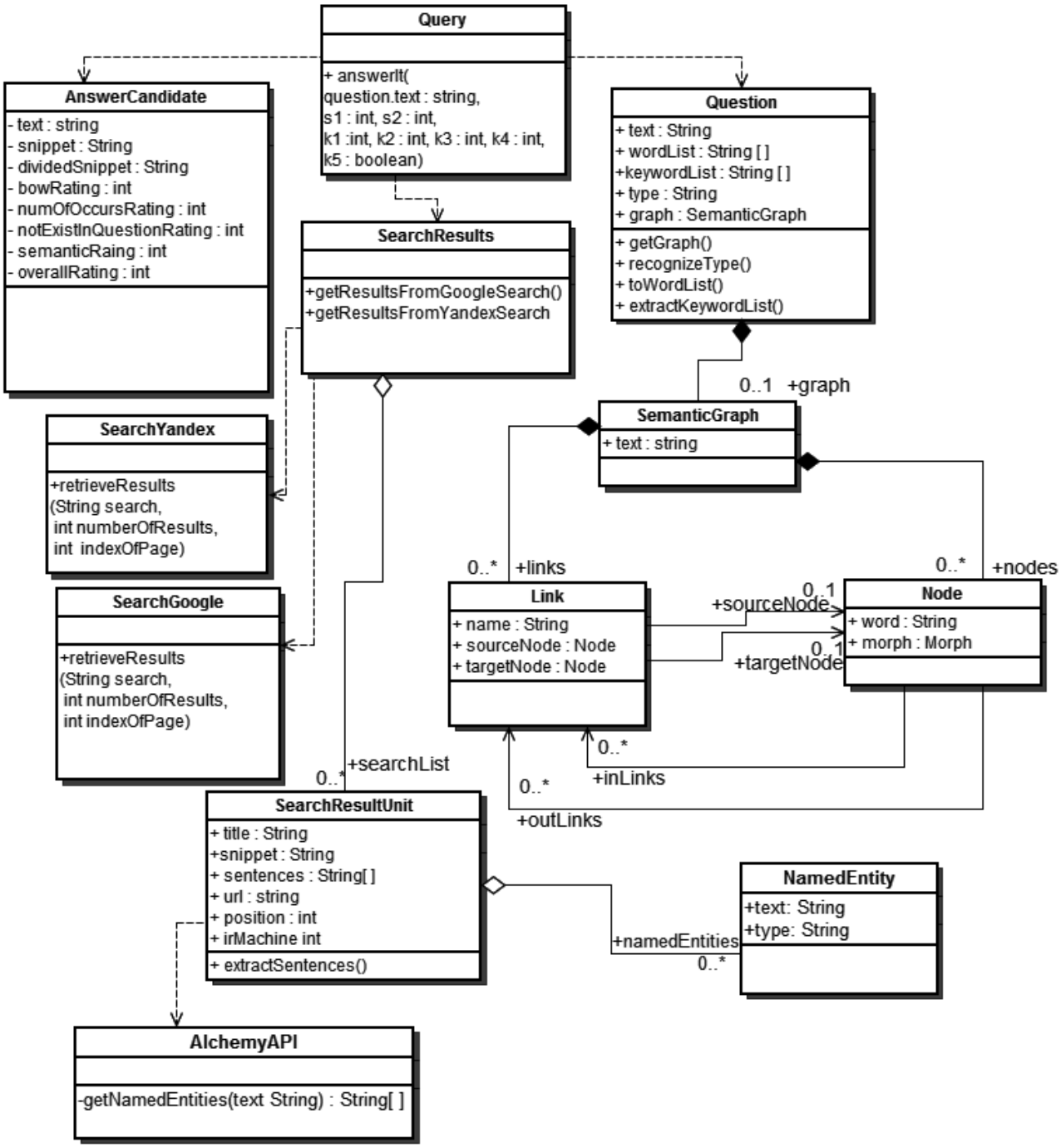

Рисунок 13. Диаграмма классов системы

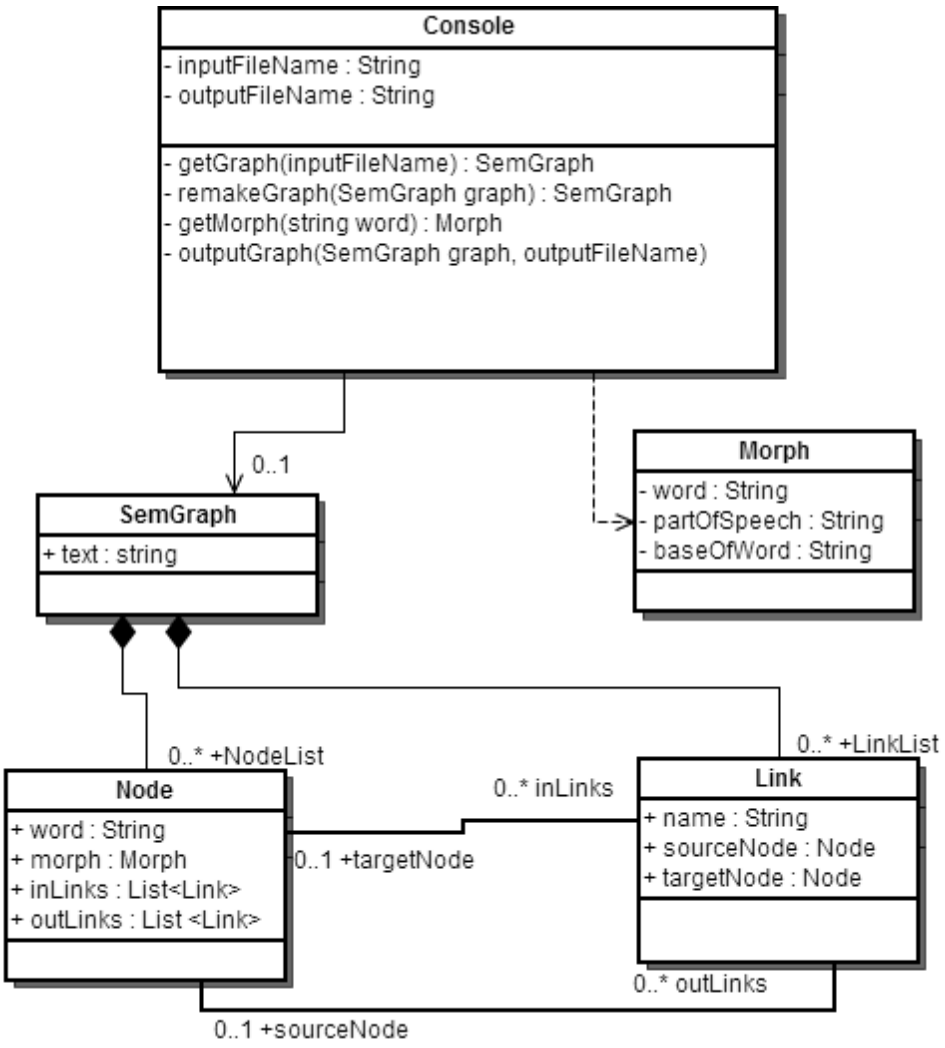

Рисунок 14. Диаграмма классов консольного приложения

Заключение

В ходе работы была исследована предметная область вопросно-ответного поиска, рассмотрены существующие современные способы решения задачи поиска ответа на вопрос и разработана модель и прототип системы для русского языка, то есть были выполнены все поставленные задачи:

1) проведение исследования предметной области вопросно-ответного поиска и существующих методов реализации вопросно-ответных систем;

2) разработка архитектуры и создание прототипа вопросно-ответной системы для русского языка, способная отвечать на вопросы об одушевлённых объектах и географических местах;

3) проведение экспериментальных исследований разработанного прототипа.

В ходе исследования был рассмотрен и реализован метод с использованием сравнения двух семантических графов предложений, предложен оригинальный метод поиска ответа на вопросы и оценки кандидата ответа с использованием семантического анализа предложений. Была реализована модель вопросно-ответной системы для русского языка без использования словаря WordNet.

Для тестирования системы был создан экспертами корпус из пар (вопрос, корректный ответ). Система была протестирована на данном корпусе.

Кроме решения поставленных задач в ходе разработки было создано приложение с графическим интерфейсом, ориентированным на пользователя.

Список литературы

1. Vanitha Guda , Suresh Kumar Sanamrudi, I.Lakshmi Manyakamba Approaches for question answering // International Journal of Engineering Science and Technology. 2011. 3. №2. P. 990-995.
2. Gaizauskas R., Humphreys K. A Combined IR/NLP Approach to Question Answering Against Large Text Collections. Department of Computer Science. Деп. в University of Sheffield, Sheffield 2010. №10.1.1.26.119.
3. Ali Mohammed Nabil Allam, Mohamed Hassan Haggag The Question Answering Systems: A Survey // International Journal of Research and Reviews in Information Sciences. 2012. 2. №3. P.10-21
4. L.Hirschman, R. Gaizauskas Natural language question answering: view from here // Natural language Engoneering. 2001. 7. №4 P. 275-300.
5. Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, Andrew Ng Web Question Answering: Is More Always Better? // SIGIR Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New-York, USA: Microsoft Research Center, 2002. P.291-298
6. Richard J Cooper, Stefan M Rüger A Simple Question Answering System // Деп. в Imperial College of Science, Technology and Medicine, 2007.
7. Аналитическая группа департамента маркетинга компании «Яндекс» Поиск в интернете: что и как ищут пользователи [PDF] (http://download.yandex.ru/company/yandex_search_mini_report_autumn_2009.pdf) .
8. Lynette Hirschman, Marc Light, Eric Breck, John D. Burger Deep Read: A Reading Comprehension System // Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. Stroudsburg,USA: Association for Computational Linguistics, 1999. P.325-333
9. D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan Building Watson: An Overview of the DeepQA Project // AI Magazine. 2010. №3. P. 59-79
10. Cheng-Lung Sung, Cheng-Wei Lee, Hsu-Chun Yen, Wen-Lian Hsu An Alignment-based Surface Pattern for a Question Answering System // Integrated Computer-Aided Engineering. 2009. 16. №3. P.259-269
11. Bernardo Magnini, Matteo Negri, Roberto Prevete, Hristo Tanev Multilingual Question Answering: the DIOGENE System // 10th Text Retrieval Conference. Italy: Centro per la Ricerca Scientifica e Tecnologica Via Sommarive, 2001. P.433-450
12. Shen Dan, Mariella Lapata Using Semantic Roles to Improve Question Answering // Joint Conference on Empirical Methods in Natural Language Processing and

Computational Natural Language Learning. Jeju, Korea: Association for Computational Linguistics, 2007. P.12-22

13. Conference and Labs Of Evaluation Forum [электронный информационный ресурс] (http://www.clef-initiative.eu).

14. Stephen Wolfram A new kind of science [HTML] (http://www.wolframscience.com/).

15. Dan Roth , Gio Kao Kao , Xin Li , Ramya Nagarajan , Vasin Punyakanok , Nick Rizzolo , Wen-tau Yih , Cecilia Ovesdotter Aim , Liam Gerard Moran Learning Components for a Question-Answering System // TREC, 2001. P. 539-548.

16. Dongli Han, Yuhei Kato, Kazuaki Takehara, Tetsuya Yamamoto, Kazunori Sugimura, Minoru Harada QA System Metis Based on Web Searching and Semantic Graph Matching // IFIP International Federation for Information Processing. 228. 2007. P.123-133

17. AOT - программное обеспечение в области автоматической обработки текста на естественном языке. [электронный ресурс] (http://www.aot.ru)

18. Punyakanok, V., Roth, D. and Yih, W. Natural language interface via dependency tree mapping: An application to question answering // AI and Math. January 2004. P.22-34.

19. S.Quateroni, S.Manandhar Designing an Interactive Open Domain Question Answering // Natural Language Engineering. 2008. №1. P.1-23.

20. Deepak Ravichandran and Eduard Hovy Learning Surface Text Patterns for a Question Answering System // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistic. Philadelphia: Information Sciences Institute University of Southern California, 2002. P. 41-47.

21. Horacio Saggion Exploring the Performance of Boolean Retrieval Strategies for Open Domain Question Answering // RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL - SIGIR, 2004

22. Sanda Harabagiu , Dan Moldovan , Marius Pasca , Rada Mihalcea , Mihai Surdeanu, Razvan Bunescu , Roxana Girju , Vasile Rus , Paul Morarescu FALCON: Boosting Knowledge for Answer Engines // Department of Science Деп. в Southern Methodist University, Dallas, 2005.

23. Dan Moldovan, Sanda Harabagiu LASSO: The structure and perfomance of open-domain question answering system // Деп.в Southern Methodist University, Dallas, 2000.

24. Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, Chin-Yew Lin Question Answering Webclopedia // Information Sciences Institute Деп. в University of Southern California

Приложение А. Графический интерфейс системы с примером её работы.