Оглавление

[Перечень условных обозначений 2](#_Toc351023399)

[Введение 3](#_Toc351023400)

[1. Общая характеристика работы 8](#_Toc351023401)

[*1.1.* *Актуальность темы исследования* 8](#_Toc351023402)

[*1.2.* *Предмет и объект* 10](#_Toc351023403)

[*1.3.* *Цель и задачи исследования* 12](#_Toc351023404)

[*1.4.* *Положения, выносимые на защиту* 12](#_Toc351023405)

[*1.5.* *Личный вклад* 13](#_Toc351023406)

[*1.6.* *Апробация результатов диссертации* 13](#_Toc351023407)

[*1.7.* *Структура и объем диссертации* 13](#_Toc351023408)

[Заключение 14](#_Toc351023409)

[Список литературы 15](#_Toc351023410)

[Приложения 16](#_Toc351023411)

# Список условных обозначений и сокращений

|  |  |  |
| --- | --- | --- |
| Условное обозначение | Расшифровка | Раздел, где вводится обозначение |
| ЕЯ(NL) | Естественный язык (Natural language) | Введение |
| ОЕЯ(NLP) | Обработка естественного языка (Natural language processing) | Введение |
| КЛ | Компьютерная лингвистика | Введение |
| CS | Computer science | Введение |
| ИИ | Искусственный интеллект | Введение |
| ВО (QA) | Вопросно-ответная(Question answering) | Введение |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Введение

Термин компьютерная лингвистика (КЛ) в последние годы все чаще встречается в связи с разработкой различных прикладных программных систем, в том числе –коммерческих программных продуктов. Связано это бурным ростом в обществе текстовой информации, в том числе в сети Интернет, и необходимостью автоматической обработки текстов на естественном языке (ЕЯ). Указанное обстоятельство стимулирует развитие компьютерной лингвистики как области науки и разработку новых информационных и лингвистических технологий.

В рамках компьютерной лингвистики, существующей уже более 50 лет (и известной также под названиями машинная лингвистика, автоматическая обработка текстов на ЕЯ) предложено много перспективных методов и идей, но далеко не все они еще нашли свое выражение в программных продуктах, используемых на практике.

Компьютерная лингвистика возникла на стыке таких наук, как лингвистика, математика, информатика (Computer Science) и искусственный интеллект. Истоки КЛ восходят к исследованиям известного американского ученого Н. Хомского в области формализации структуры естественного языка; ее развитие опирается на результаты в области общей лингвистики (языкознания). Языкознание изучает общие законы естественного языка – его структуру и функционирование, и включает такие области:

* *Фонология* – изучает звуки речи и правила их соединения при формировании речи;
* *Морфология* – занимается внутренней структурой и внешней формой слов речи, включая части речи и их категории;
* *Синтаксис* – изучает структуру предложений, правила сочетаемости и порядка следования слов в предложении, а также общие его свойства как единицы языка.
* *Семантика и прагматика* – тесно связанные области: семантика занимается смыслом слов, предложений и других единиц речи, а прагматика – особенностями выражения этого смысла в связи с конкретными целями общения;
* *Лексикография* описывает лексикон конкретного ЕЯ – его отдельные слова и их грамматические свойства, а также методы создания словарей.

КЛ тесно связана и с такой междисциплинарной научной областью, как искусственный интеллект (ИИ), в рамках которого разрабатываются компьютерные модели отдельных интеллектуальных функций. Одна из первых работающих программ в области ИИ и КЛ – это известная программа Т. Винограда, которая понимала простейшие приказы человека по изменению мира кубиков, сформулированные на ограниченном подмножестве ЕЯ. Отметим, что несмотря на очевидное пересечение исследований в области КЛ и ИИ (поскольку владение языком относится к интеллектуальным функциям), ИИ не поглощает всю КЛ, поскольку она имеет свой теоретический базис и методологию. Общим для указанных наук является компьютерное моделирование как основной метод и итоговая цель исследований.

Область приложений компьютерной лингвистики постоянно расширяется. Одно из относительно новых направлений связанных с компьютерной лингвистикой-формирование ответов на вопросы или вопросно-ответная система (Question Answering). Эта задача решается путем определения типа вопроса, поиском текстов, потенциально содержащих ответ на этот вопрос, и извлечением ответа из этих текстов.

С момента появления первых прототипов вопросно-ответных систем их область применения значительно расширилась. Например, их используют в ответах на вопросы, связанные со временем, геолокационные вопросы, вопросы определения понятий, библиографические, многоязыковые вопросы, вопросы, связанные с мультимедиа (визуальной, аудио- и видео- информацией). Изучаются смежные области, такие как построение интерактивных QA-систем (уточняющие вопросы, требующиеся для разъяснения первоначального), повторное использование ответов и представление знаний, использование логического вывода из имеющей информации для получения ответов на вопросы и т. п., прогнозирование, какие вопросы могут быть заданы, анализ настроения.

# Понятие вопросно-ответной системы

Вопросно-ответная система – это информационная система, являющаяся гибридом поисковых, справочных и интеллектуальных систем, которая использует естественно-языковой интерфейс. На вход такой системе подаётся запрос, сформулированный на естественном языке, после чего он обрабатывается с использованием методов NLP, и генерируется естественно-языковой ответ. В качестве базового подхода к задаче поиска ответа на вопрос обычно применяется следующая схема: сначала система тем или иным образом (например, поиском по ключевым словам) отбирает документы, содержащие информацию, связанные с поставленным вопросом, затем фильтрует их, выделяя отдельные текстовые фрагменты, потенциально содержащие ответ, после чего из отобранных фрагментов генерирующий модуль синтезирует ответ на вопрос.

* 1. **Анализ существующих работ**

Первые QA-системы были разработаны в 1960х годах и являлись естественно-языковыми оболочками для экспертных систем, ориентированных на конкретные области. Среди наиболее известных реализаций следует выделить системы BASEBALL и LUNAR. Система BASEBALL позволяла вести диалог с пользователем, интересующимся результатами соревнований бейсбольной лиги США за прошедший год. Система LUNAR отвечала на вопросы, связанные с геологическим анализом образцов пород, доставленных с лунной поверхности экспедициями программы «Аполлон». Обе системы были остаточно эффективно реализованы и представляли собой примеры реализации вопросно-ответных систем, ориентированных на конкретную предметную область. Например, система LUNAR была продемонстрирована на конференции 1971 года, на которой обсуждались вопросы лунных исследований, позволяла получить ответы на порядка 90% всех вопросов, заданных данной системе.

Некоторые известные программные системы, разработанные в 60-х

годах прошлого века, также содержали в себе вопросно-ответные модули

в виде подсистем. Например, программа ELIZA1 [1] содержала в качестве

программного модуля вопросно-ответную систему, которая, собственно,

и позволяла общаться с пользователем.

В 70-х и 80-х годах прошлого века было реализовано достаточно много вопросно-ответных систем, позволяющих вести диалог с пользователем в конкретной предметной области. Например, программный комплекс Unix Consultant отвечал на вопросы, связанные с операционной системой UNIX. Unix Consultant был основан на достаточно сложной и развитой базе знаний, содержащей информацию об операционной системе UNIX. Интерфейс к базе знаний был реализован в виде вопросно-ответной системы.

Все рассмотренные реализации вопросно-ответных систем позволяли отвечать на вопросы, связанные с конкретной предметной областью—

узкоспециализированных вопросно-ответных систем. В конце 90-х годов

прошлого века, в связи с развитием Интернета и Веб, была осознана необходимость создания вопросно-ответных систем, не связанных с какой-либо предметной областью—т.н. открытых вопросно-ответных систем. Такие системы позволяют вести диалог по всем областям знаний, например, на основе частично структурированных знаний, содержащихся в Веб.

В настоящее время имеется довольно большое количество реализаций

вопросно-ответных систем. Заслуживает внимания реализация вопросно-ответной системы START [2]. Хороший обзор этой системы на русском языке был дан в [3].

Другой интересной системой, претендующий на статус открытой, является система Cyc [4]. История создания и развития этой системы насчитывает уже порядка 30 лет. За это время в системе разработано множество онтологий, как говорится, на все случаи жизни. Изначально

представляющая собой объемную базу знаний, систем Cyc предоставляет также интерфейс на английском языке и, таким образом, является вопросно-ответной системой. Для более детальной информации о системе Cyc см. [5, 6].

С 1999 года проводится соревнование по вопросно-ответным системам в рамках конференции TREC, с 2003 года соревнования вопросно-ответных систем в многоязычном контексте начаты на конференции CLEF. Современные системы предназначаются для поиска ответов на вопросы в предоставляемых документах с использованием технологий обработки естественных языков или natural language processing (NLP).

Современные QA-системы обычно включают особый модуль — классификатор вопросов, который определяет тип вопроса и, соответственно, ожидаемого ответа. После этого анализа система постепенно применяет к предоставленным документам все более сложные и тонкие методы NLP, отбрасывая ненужную информацию. Самый грубый метод — поиск в документах — предполагает использование системы поиска информации для отбора частей текста, потенциально содержащих ответ. Затем фильтр выделяет фразы, похожие на ожидаемый ответ (например, на вопрос «Кто …» фильтр вернет кусочки текста, содержащие имена людей). И, наконец, модуль выделения ответов найдет среди этих фраз правильный ответ.

Таким образом, задача QA системы это ответить на вопросы, в результате поиска в Web или в других коллекциях документов небольших сегментов текста, содержащие ответы к вопросам, их переформатирование и представление пользователю (см. пример 1).

|  |  |
| --- | --- |
| Вопрос | Ответ |
| Когда родился Платон? | 428 или 427 г. До н.э. |
| Кто был учеником Сократа? | Платон |

Пр. 1. Возможные вопросы и ответы на них.

QA системы можно условно разделить на:

* *Узкоспециализированные QA*-системы работают в конкретных областях (например, медицина или обслуживание автомобилей). На данный момент самый привлекательным кажется проект компании IBM “Watson”.
* *Общие QA*-системы работают с информацией по всем областям знаний, таким образом появляется возможность вести поиск в смежных областях.
  1. *Актуальность темы исследования*

Актуальностью данных может служить тот факт, что в связи с непрерывным увеличением объемов информации, доступной в глобальной сети интернет необходим более эффективный поиск и доступ к данным, чем стандартный поиск по ключевым словам, так как он не учитывает языковые и смысловые связи между словами запроса.

*Проблемы ВО*

В 2002 году группа исследователей написала план исследований в области вопросно-ответных систем. Предлагалось рассмотреть следующие вопросы.

1. Типы вопросов

Разные вопросы требуют разных методов поиска ответов. Поэтому нужно составить или улучшить методические списки типов возможных вопросов.

1. Обработка вопросов

Одну и ту же информацию можно запросить разными способами. Требуется создать эффективные методы понимания и обработки семантики (смысла) предложения. Важно, чтобы программа распознавала эквивалентные по смыслу вопросы, независимо от используемых стиля, слов, синтаксических взаимосвязей и идиом. Хотелось бы, чтобы QA-система разделяла сложные вопросы на несколько простых, и правильно трактовала контекстно-зависимые фразы, возможно, уточняя их у пользователя в процессе диалога.

1. Контекстные вопросы

Вопросы задаются в определенном контексте. Контекст может уточнить запрос, устранить двусмысленность или следить за ходом мыслей пользователя по серии вопросов.

1. Источники знаний для QA-системы

Перед тем как отвечать на вопрос, неплохо было бы осведомиться о доступных базах текстов. Какие бы способы обработки текстов не применялись, мы не найдем правильного ответа, если его нет в базах.

1. Выделение ответов

Правильное выполнение этой процедуры зависит от сложности вопроса, его типа, контекста, качества доступных текстов, метода поиска и др. — огромного числа факторов. Поэтому подходить к изучению методов обработки текста нужно со всей осторожностью, и эта проблема заслуживает особого внимания.

1. Формулировка ответа

Ответ должен быть как можно более естественным. В некоторых случаях достаточно и простого выделения его из текста. К примеру, если требуется наименование (имя человека, название прибора, болезни), величина (денежный курс, длина, размер) или дата («Когда родился Иван Грозный?») — прямого ответа достаточно. Но иногда приходится иметь дело со сложными запросами, и здесь нужны особые алгоритмы слияния ответов из разных документов.

1. Ответы на вопросы в реальном времени

Нужно сделать систему, которая бы находила ответы в хранилищах за несколько секунд, независимо от сложности и двусмысленности вопроса, размера и обширности документной базы.

1. Многоязыковые запросы

Разработка систем для работы и поиска на других языках (в том числе автоматический перевод).

1. Интерактивность

Зачастую информация, предлагаемая QA-системой в качестве ответа, неполна. Возможно, система неправильно определила тип вопроса или неправильно «поняла» его. В этом случае пользователь может захотеть не только переформулировать свой запрос, но и «объясниться» с программой с помощью диалога.

1. Механизм рассуждений (вывода)

Некоторые пользователи хотели бы получить ответ, выходящий за рамки доступных текстов. Для этого в QA-систему нужно добавить знания, общие для большинства областей (см. Общие онтологии в информатике), а также средства автоматического вывода новых знаний.

1. Профили пользователей QA-систем

Сведения о пользователе, такие как область интересов, манера его речи и рассуждения, подразумеваемые по умолчанию факты, могли бы существенно увеличить производительность системы.

* 1. *Предмет и объект*

Вопросно-ответная система состоит из трех ключевых стадий: обработка вопроса, извлечение или поисковый проход, обработка ответа (см. рис. 1).

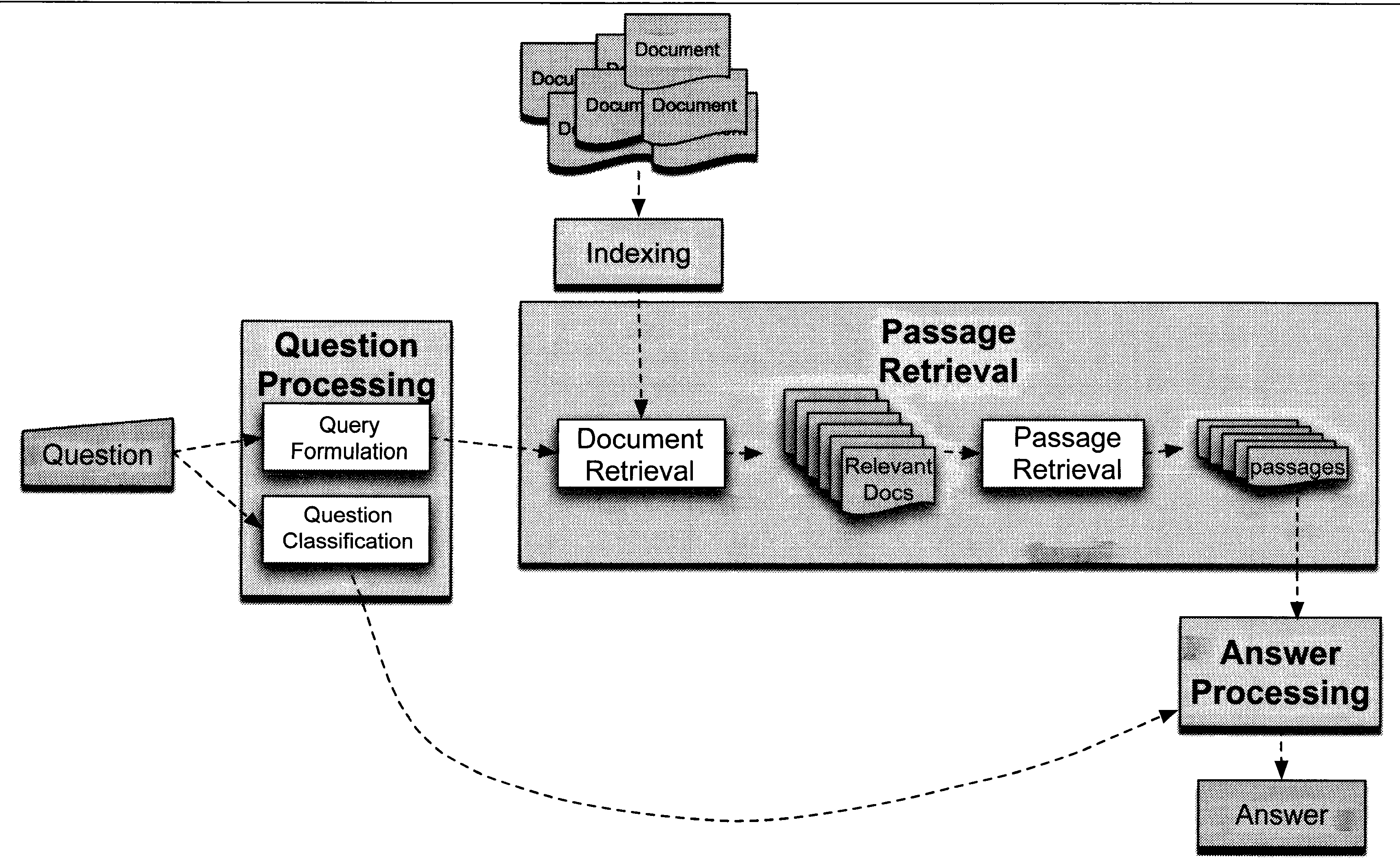


Рис. 1. Архитектура вопросно-ответной системы.

***Обработка вопроса.***

Цель этой фазы заключается в извлечении двух положений: ключевой запрос подходящий для системы информационного поиска (IR) и тип ответа, специфицирующий по типу объекта, который будет возвращать в качестве ответа, то есть классификация вопроса.

***Формулировка запроса.***

Процесс формулирования запроса аналогичен обработке другого IR запроса. Цель является создать из вопроса список ключевых слов для IR запроса. То есть необходимо извлечь ключевые термины из вопроса, также возможно расширить вопрос лексически/семантически близкими словами. Таким образом, вопрос можно смоделировать как множество ключевых слов (пр. 2).

|  |  |
| --- | --- |
| Вопрос | Ключевые слова |
| С чем сравнивает Платон чувственное познание? | Платон, сравнение, чувство, познание |
| Кому из учеников Платона приписываются слова: «Платон мне друг, но истина дороже!»? | Ученик, Платон, слова, истина |

Пр. 2. Моделирование с помощью ключевых слов.

Также возможно построение ответа на основе переформулирования запроса. Для этого применяется следующие правила:

• Приведение к форме подстроки декларативного ответа («когда был придуман лазер?»-«лазер был придуман…»)

• Послать переформулированный запрос информационно-поисковой системе

• Применение более общего правила (wh-word did A verb B->…A verb+ed B, Where is A-> A is located in…)

***Классификация вопроса.***

Второй задачей обработки запроса является классификация вопроса и определение типа ответа. Например, такой вопрос как «Кто написал произведение ″Апология Сократа″»? предполагает ответ типа ″Человек″. Вопрос «Какой самый большой город в России?» предполагает ответ типа ″Город″.

Классификатор вопросов может быть построен с помощью правил составленных вручную, машинное обучение с учителем или их комбинация. Например, ″The Webclopedia QA Typology″ содержит 276 размеченных вручную правил, связанных со 180 приближенных типов ответа в типологии.

Однако самые современные классификаторы вопросов основаны на методах обучения с учителем. Эти классификаторы тренируют на базе данных вопросов, размеченных на таких корпусах как Ли и Рота (Li and Roth(2002)).

***Извлечение фрагментов текста.***

Во время следующей стадии обработки запроса IR-система должна возвратить список релевантных документов, в которых встречаются те или иные ключевые слова. Стоит отметить, что необходимым фрагментов может быть как предложение, так и целый параграф. Но, если отметить специфику нашей базы данных, то возвращаемым документов может быть только предложение или набор предложений. Таким образом, необходимо выбрать фрагменты, потенциально содержащие ответ. Для этого нужно отсеять фрагменты, не содержащие ответ, то есть распознать именованные сущности и классифицировать ответ, потом ″отранжировать″ оставшиеся фрагменты согласно либо правилам, составленным вручную, либо с помощью машинного обучения. Если извлечение фрагментов происходит из Интернета, то можно пропустить шаг извлечения фрагмента и использовать сниппеты, возвращаемые информационно-поисковыми системами.

***Обработка ответа.***

Последней стадией работы QA-системы является извлечение конкретного ответа из IR системы и представление его пользователю как ″300 миллионов″ на вопрос ″Каково население США?″.

Существуют два класса алгоритмов для задачи извлечения ответа:

* основанный на шаблоне;
* сбор из N-грамм (N-gram tiling).

В алгоритме на основе шаблона используется информация о типе в регулярных выражениях (например, если тип ответа ″Человек″, то извлечь именованные сущности ″Человек″ из фрагмента). Но при этом некоторые типы ответов (например, определения) не подразумевают конкретного типа именованной сущности в ответе. Для этого используют регулярные выражения (созданные вручную или автоматически) (пр.3).

|  |  |  |
| --- | --- | --- |
| Образец | Вопрос | Ответ |
| <ОФ> such as <ВФ> | What is autism | ″,developmental disorders such as autism″ |

Альтернативный метод, N-gramm tiling, представлен на рис.2.



Рис. 2. Архитектура сбора ответа из N-gram.

На первом шаге происходит выявление N-gram, то есть каждую юниграму, биграму и триграму, встречающуюся в сниппете, извлекают и ″взвешивают″. Весом является функция от числа сниппетов а которых встречается N-gram’а. На следующем шаге, фильтрации N-gram, происходит ″перезвешивание″ с учетом типа ответа. И, наконец, алгоритм сбора N-gram объединяет перекрывающиеся фрагменты N-gram в ответе. Стандартный «жадный» алгоритм начинает с самого «тяжелого» кандидата и повторяет сцепление фрагмента с другими кандидатами. Объединение с самой высокой оценкой добавляется в множество кандидатов, кандидаты с низкой оценкой удаляются и процесс продолжается пока не будет составлен один ответ.

технологий обработки естественных языков (NLP).

* 1. *Цель и задачи исследования*
  2. *Положения, выносимые на защиту*
  3. *Личный вклад*
  4. *Апробация результатов диссертации*
  5. *Структура и объем диссертации*

# Заключение

# Список литературы

1. Статья о программе ELIZA на сайте Википедия. URL: http://en.wikipedia.org/wiki/ELIZA (дата обращения: 07.03.2012).

2. Сайт проекта START. URL: http://start.csail.mit.edu (дата обращения: 11.03.2012).

3. Никитин А., Райков П. Вопросно-ответные системы. URL: yury.name/internet/06ia-seminar.ppt (дата обращения: 11.03.2012).

4. Сайт проекта Cyc. URL: http://www.cyc.com (дата обращения: 11.03.2012).

5. Лапшин В. Система Cyc и ее библиотека онтологий (часть первая) // Искусственный интеллект и принятие решений. 2010. № 2.

6. Лапшин В. Система Cyc и ее библиотека онтологий (часть вторая) // Искусственный интеллект и принятие решений. 2010. № 3.

7. К.Д. Манниг, Х. Шютце, П. Рагхаван. Введение в информационный поиск.

8. D. Jurafsky, J. H. Martin. Speech and language processing.

# Приложения