

Программная реализация на базе платформы Apache Jena вопросно-ответной системы, использующей данные онтологий

А.В. Мочалова, В.А. Мочалов

Институт космофизических исследований и распространения радиоволн ДВО РАН

stark345@gmail.com, sensorlife@mail.ru

Аннотация

В настоящее время весьма актуальной задачей является разработка вопросно-ответных систем, позволяющих отвечать на вопросы пользователей, заданные на естественном языке по машиночитаемым текстам на естественном языке.

В работе описывается вопросно-ответная система, использующая данные из онтологии RuTez.

Предлагаются этапы решения задачи соотнесения частей текста с узлами онтологии. Всего выделяется 6 этапов: предварительная обработка текста; определение границ предложений; выделение границ синтаксем; определение возможных вариантов лемм для всех выделенных синтаксем; поиск в онтологии элементов, соответствующих начальным формам синтаксем; выбор из элементов онтологии, соответствующих синтаксемам.

Приводится описание архитектуры вопросно-ответной системы, основанной на использовании платформы Apache Jena, экспертной системы Drools и разработанного авторами семантического анализатора. Приводятся сквозные примеры работы системы.

Ключевые слова: вопросно-ответные системы, онтологии, автоматическая обработка текста, Drools, SPARQL, Apache Jena

1. Введение

1.1. Общая схема работы вопросно-ответной системы

На основе анализа существующих разработок вопросно-ответных систем, можно сделать вывод о том, что качественная система ответа на вопрос функционирует в соответствии с определенной схемой, представленной на рисунке 1 (подробный анализ вопросно-ответных систем проведен в работе [1]). На вход системе подается вопрос, сформулированный на естественном языке. Затем текст вопроса проходит автоматическую обработку, основные этапы которой следующие: предварительная обработка текста (включает в себя удаление лишних символов форматирования, исправление орфографических и пунктуационных ошибок, удаление лишних пробелов и символов переноса строк и т.п.); извлечение именованных сущностей; разбиение текста на предложения; токенизация (разбиение предложений на слова); морфологический, синтаксический и семантический анализ. Модуль автоматического анализа текста, как правило, использует различные структурированные лингвистические ресурсы: словари, базы данных, базы фактов, онтологии. В некоторых вопросно-ответных системах часть из вышеперечисленных этапов автоматической обработки текстов может быть пропущена или выполняться в упрощенном виде, а часть наоборот — являться основополагающими для

работы всей системы, как, например, семантический анализ в работе М.В. Мозгового [2]. Затем текст вопроса классифицируется в соответствии с принятой в данной системе классификацией. На базе результатов автоматической обработки текста вопроса и результатов классификации вопроса формируется запрос, который передается поисковой машине.

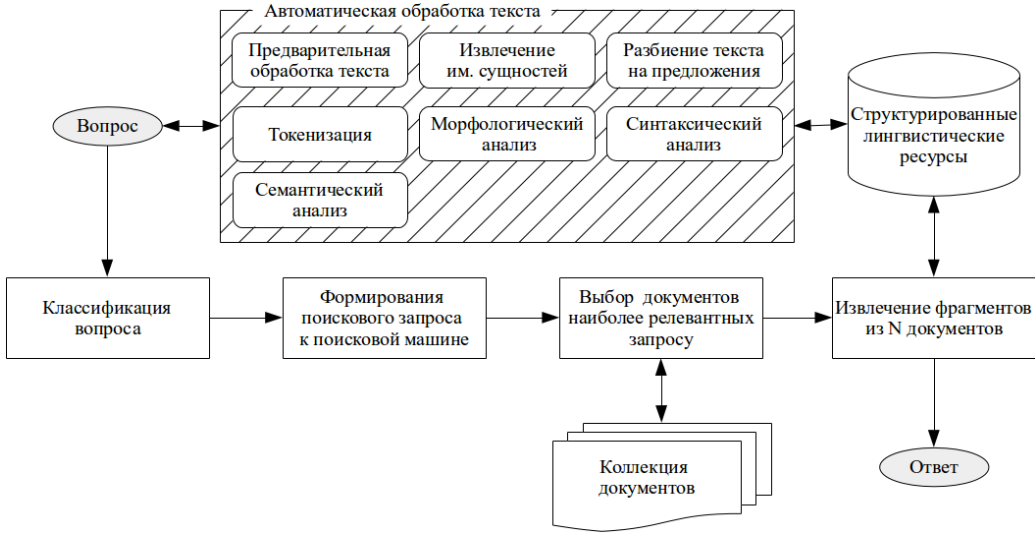


Рис. 1. Схема работы вопросно-ответной системы

Далее поисковая машина выбирает определенное количество документов (если поиск производится не по одному заданному документу), наиболее релевантных запросу. Выбор документов может производиться с помощью внешних поисковых систем, либо с помощью собственной поисковой машины, являющейся частью разрабатываемой системы. В некоторых случаях поиск документов производится в ограниченной специализированной коллекции документов, которой располагает система. Эффективный подход к организации поисковой системы предложен в работе [3], где предлагается архитектура субпоисковой системы, которая формирует собственную базу документов и собственный поисковый индекс, а для ускорения процесса сбора потенциально интересующих документов использует внешние поисковые системы (Google, Яндекс, Bing). На рисунке 2 представлена диаграмма соотношений множеств документов в такой субпоисковой системе (I — множество документов, доступных в сети интернет, W — множество документов, отобранных Интернет поисковой системой, S — множество документов, отобранных субпоисковой системой) [3].

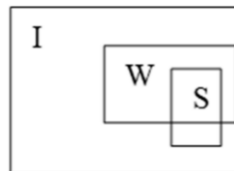


Рис. 2. Диаграмма соотношений множеств документов в субпоисковой системе

Текст каждого из выбранных документов, также как и текст вопроса, подвергается автоматической обработке. При этом алгоритмы машинной обработки текста вопроса могут отличаться от алгоритмов машинной обработки набора документов, выбранных поисковой

системой. Далее, посредством внутренних алгоритмов работы вопросно-ответной системы, происходит выбор конкретных фрагментов текстов из документов, переданных поисковой системой. Выбранные фрагменты текста представляются системой в качестве ответа. Наиболее продвинутые вопросно-ответные системы на этапе выбора фрагментов текста могут использовать данные из структурированных лингвистических ресурсов. Информация из этих лингвистических ресурсов может дополнять ответ/ответы системы.

В связи с тем, что результаты машинной обработки текста передаются модулям вопросно-ответной системы, от которых напрямую зависит ответ системы на вопрос, можно сделать вывод, что задача автоматической обработки текста является одной из важнейших задач, решаемых в рамках работы вопросно-ответной системы и от корректности работы модуля обработки текста напрямую зависит корректность работы всей системы.

1.2. Применение тезаурусов при разработке вопросно-ответных систем

Одно из пониманий тезауруса подразумевает словарь, с максимальной полнотой представляющий лексику языка во всей ее полноте с примерами употребления в текстах. Однако с точки зрения применения тезауруса в вопросно-ответной системе его следует понимать как информационно-поисковый тезаурус, как словарь общей или чаще специальной лексики, в котором в явном виде указаны семантические отношения между лексическими единицами (синонимия, антонимия, гипонимия, гиперонимия и т.п.). Многие прочие отношения часто объединяются в общий класс ассоциативных отношений. В отличие от толкового словаря, тезаурус позволяет выявлять смысл не только с помощью определения, но и посредством соотнесения слова с другими понятиями и их группами. Тезаурус — это терминологический ресурс, реализованный в виде словаря понятий и терминов со связями между ними. Основное его назначение — помощь при информационном поиске: на основе связей тезауруса происходит расширение запроса, навигация по связям тезауруса помогает четче сформулировать сам запрос.[4]

Качество работы вопросно-ответной системы напрямую зависит от качества и объема используемых тезаурусов. Использоваться они могут на разных уровнях реализации такой системы, например, при определении границ синтаксем, выделении именованных сущностей, в модуле, выполняющем семантический анализ текста, а также непосредственно в алгоритмах поиска ответа на вопрос в анализируемом тексте.

Следует отметить, что в широком понимании онтологии тезаурусы тоже являются онтологиями.

В этой работе описывается вопросно-ответная система, использующая данные из известного лингвистического ресурса — тезауруса RuТез [5], хранящего данные в структурированном виде. Описывается использование RuТез для выделения именованных сущностей в тексте, показывается как с помощью SPARQL-запросов и онтосемантического анализатора, используемого вопросно-ответной системой, формируются ответы на заданные пользователями вопросы.

2. Архитектура вопросно-ответной системы на базе платформы Apache Jena

В настоящее время весьма актуальной задачей является разработка вопросно-ответных систем, позволяющих отвечать на вопросы пользователей, заданные на естественном языке по машиночитаемым текстам на естественном языке. Ниже приводится описание архитектуры вопросно-ответной системы, основанной на базе платформы Apache Jena и использующей данные из онтологии [1].

На рисунке 3 приводится обобщенная архитектура вопросно-ответной системы [6], основанная на использовании семантического анализатора, построенного по математической модели, описанной в работе [7].

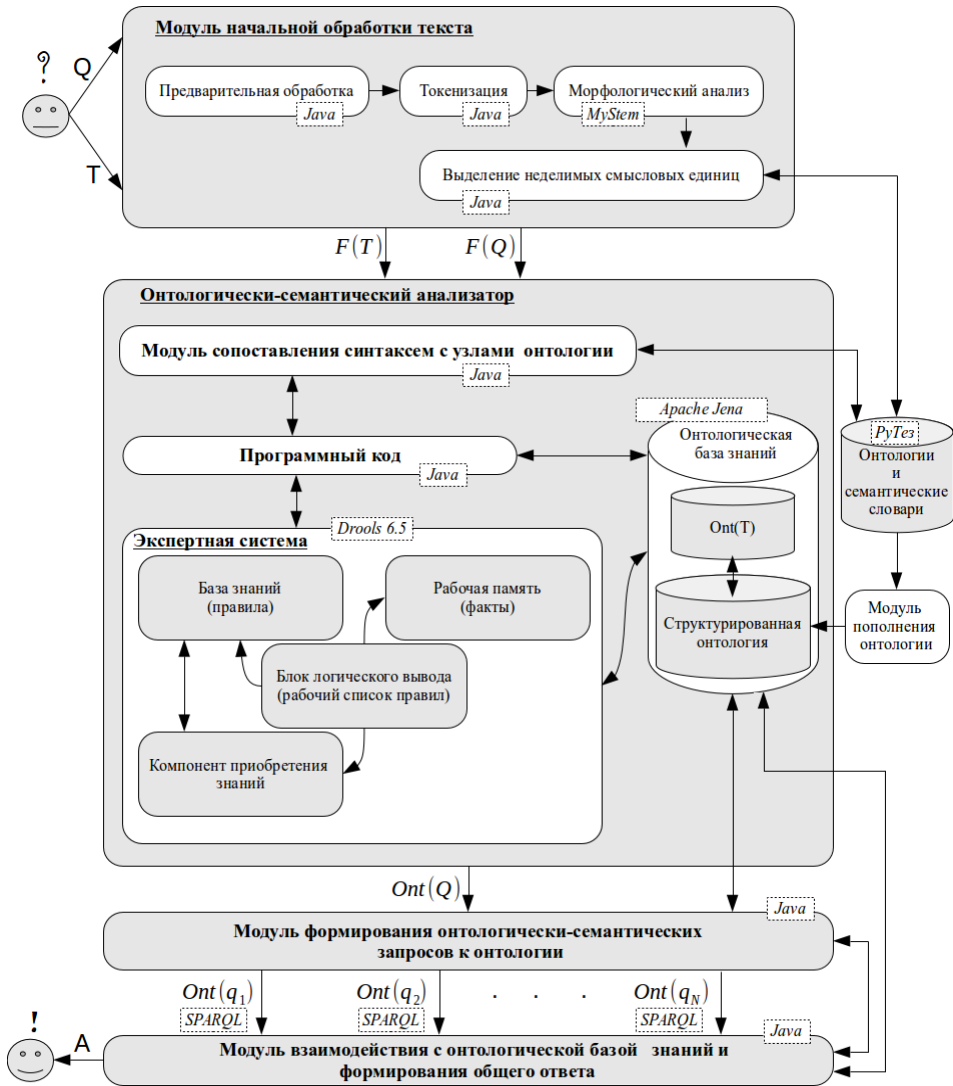


Рис. 3. Пример архитектуры вопросно-ответной системы

На вход вопросно-ответной системе подается вопрос Q на естественном языке и пользовательский текст T , выбранный пользователем для поиска ответа на вопрос Q . Текст T и вопрос Q поступают на вход модуля начальной обработки текста, в котором происходит выполнение следующих шагов: предварительная обработка, токенизация, морфологический анализ, выделение неделимых смысловых единиц. Результаты начальной обработки T и Q записываются в $F(T)$ и $F(Q)$ соответственно, после чего $F(T)$ и $F(Q)$ поступают на вход онтологически-семантическому анализатору, который на основе использования программного кода, экспертной системы, онтологической базы знаний выполняет следующие действия: сопоставление синтаксис с узлами структурированной онтологии; построение онтосемантических графов $Ont(T)$ и $Ont(Q)$, узлы которого ссылаются на элементы структурированной онтологии. Структурированная онтология формируется на базе загружаемых в систему онтологий и семантических словарей с помощью модуля пополнения онтологии. Далее $Ont(Q)$ подается на вход модуля формирования онтологически-семантических запросов $Ont(q_i)$ к онтологии, которые

отправляются модулю взаимодействия с онтологией и формирования общего ответа. Этот модуль выполняет запросы к онтологической базе знаний и на базе полученных ответов формирует общий ответ А.

Пунктирной рамкой на рисунке 3 обведены названия языка программирования (Java), экспертной системы (Drools [8]) онтологии (PyTез [5]), семантической платформы (Apache Jena [9]) и языка запросов (SPARQL [10]), с помощью которых был программно реализован прототип такой системы.

Идея использования SPARQL-запросов при разработке вопросно-ответных систем не нова: пример такой системы описывается в работе [11].

3. Соотнесение частей текста с узлами онтологии

Необходимость определения соответствия частей текста и элементов онтологии возникает при решении целого ряда задач компьютерной лингвистики, связанных с автоматической обработкой текста (например, при реализации систем машинного перевода, автоматического аннотирования и реферирования, при разработке информационно-поисковых и вопросно-ответных систем, систем разметки корпусов текста и др.).

При решении задачи соотнесения частей текста с узлами онтологии можно выделить следующие этапы:

- (s1) Предварительная обработка текста;
- (s2) Определение границ предложений;
- (s3) Выделение границ синтаксем;
- (s4) Определение возможных вариантов лемм для всех выделенных синтаксем;
- (s5) Поиск в онтологии элементов, соответствующих леммам из (s4);
- (s6) Выбор из элементов онтологии, найденных в (s5), тех, которые соответствуют синтаксемам из (s3).

Первый этап — «Предварительная обработка текста» может включать такие действия по обработке естественно-языкового текста, представленного в электронном виде, как удаление символов форматирования текста, удаление лишних пробелов и переносов строк, исправление опечаток, правка всевозможных машинно-определяемых ошибок в написании оформлении текста.

Ниже приведены наиболее известные программные реализации, выполняющие некоторые задачи предварительной обработки текстов с указанием названия и вида лицензии для каждой из них или условий использования (приведено в скобках).

Проверка правописания:

- GNU Aspell (LGPL),
- Hunspell (GPL, LGPL, MPL),
- ОРФО Speller (Коммерческая),
- ОРФО Grammar Checker (Коммерческая).
- Проверка грамматики: LanguageTool (LGPL), Microsoft Word (Коммерческая).

Для решения второго этапа «Определение границ предложений» в сети Internet предлагается множество программных реализаций, выполняющих такую разбивку. Однако, в основе работы большинства таких программ лежит принцип определение конца предложения по терминальному знаку препинания (точка, вопросительный или восклицательный знак). Такой подход к решению задачи сегментации предложений привлекает своей простотой, но в реальной программной системе соотнесения частей текста с узлами онтологии использовать его нежелательно т.к. количество ошибочно найденных границ предложения при использовании описанного подхода, неоправданно велико.

В отечественной литературе проблема разбиения русскоязычного текста на предложения кратко освещается в работе [12]. В работе [13] предлагается метод автоматической сегментации русскоязычного текста на предложения на основе анализа контекста потенциальных границ предложений, при этом потенциальные границы

определяются либо посредством терминальных знаков, либо вообще с помощью всей пунктуация. При этом авторы не рассматривают предложения, не заканчивающиеся никаким знаком препинания.

Среди наиболее известных программных реализаций, выполняющих разбиение русскоязычного текста на предложения, можно выделить Aot (лицензия LGPL) — как часть графематического анализа и RCO (коммерческая лицензия).

Третий этап «Выделение границ синтаксем» — однозначно, самая сложная из задач, предшествующих непосредственно решению задачи соотнесения частей текста с узлами онтологии. Под синтаксемой будем понимать единицу текста, которая в работе [14] определяется как минимальная, далее неделимая семантико-синтаксическая единица русского языка, выступающая одновременно как носитель элементарного (категориально-семантического) смысла и как конструктивный элемент более сложных синтаксических построений. От того насколько корректно будут определены границы синтаксем в анализируемом тексте, напрямую зависит качество работы системы соотнесения частей этого текста с узлами онтологии.

Для примера рассмотрим предложение «В лесу у моря стоит замок» (см. рис. 4). Поставив задачей определить соответствие частей этого текста с элементами Wikidata — базы данных, которую также можно классифицировать как онтологию, столкнемся с трудностями, связанными с определением границ синтаксем. Например, в Wikidata присутствует как элемент «лес», имеющий несколько различных значений, так и элемент «В лесу», характеризуемого как «рассказ Бориса Александровича Лазаревского». Очевидно, что в зависимости от того как будут определены границы синтаксем в анализируемом предложении, напрямую зависит корректность соотнесения частей этого предложения с элементами онтологии.

После того, как в анализируемом тексте определены границы синтаксем, необходимо определить леммы (начальные формы) для всех синтаксем т.к. элементы онтологии обычно хранятся в начальной форме. Здесь начинается четвертый этап решения задачи соотнесения частей текста с элементами грамматический словарь Зализняка [15]. Для рассматриваемого в примере предложения «В лесу у моря стоит замок» с помощью словаря Зализняка для 3 синтаксем из 6 будут определены 2 леммы: слову «лесу» соответствуют леммы «леса» и «лес», слову «моря» — леммы «море» и «морить», слову «стоит» — леммы «стоять» и «стоять». Остальные синтаксемы анализируемого предложения употреблены в формах, совпадающих с их леммами (см. рис. 4).

Далее следует этап поиск в онтологии элементов, соответствующих леммам, найденным на предыдущем этапе. Для рассматриваемого в примере предложения для 7 лемм из 9 в онтологии РуТез будут найдены узлы с именами этих лемм. Ниже перечислены значения этих лемм (в соответствии с РуТез):

- Леса → {{рыболовная леса}};
- Лес → {{деловая древесина}; [лесной массив]; [лес (множество чего-н. поднятого)}};
- Море → {{водный объект}; [море (большое количество)}};
- Морить → {{травить отравой}; [морить (мучить, изнурять)]; [морение древесины}};
- Стоять → {{стоять (быть без движения)]; [стоять (бездействовать)]; [находиться, пребывать]; [стоять (сохраняться, не портиться)]; [стоять в вертикальном положении]};
- Стоить → {{подобать, надлежать, следовать}; [стоять, иметь цену]};
- Замок → {{замок для запираания}; [средневековый замок]}.

Для синтаксем «в» и «у» в онтологии РуТез не найдено элементов с названиями лемм этих синтаксем.

На завершающем этапе требуется из элементов онтологии, найденных для всех возможных лемм каждой синтаксемы, выбрать единственный. Для примера, предложения, рассматриваемого ранее, синтаксеме «лесу» должен быть поставлен в соответствие элемент онтологии «лесной массив», синтаксеме «моря» — элемент «водный объект», синтаксеме «стоит» — элемент «находиться, пребывать», и синтаксеме «замок» — элемент онтологии

«средневековый замок». На рисунке 4 выбранные элементы онтологии RuTез выделены серым цветом.

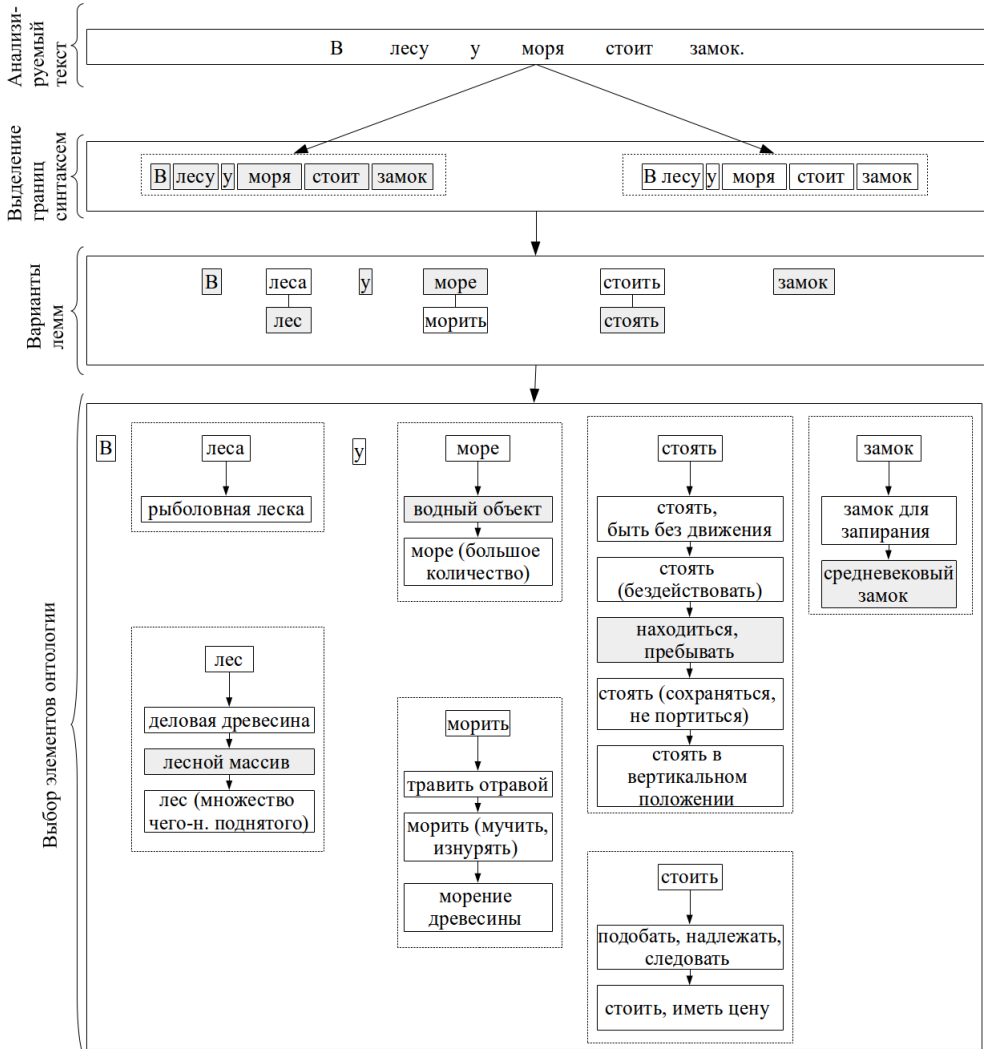


Рис. 4. Некоторые этапы решения задачи соотнесения частей текста с узлами онтологии

Краткий обзор методов и алгоритмов разрешения лексической многозначности приведен в работе [16]. Среди подходов к разрешению лексической многозначности выделяют методы, основанные на использовании внешних источников информации и методы, основанные на машинном обучении (обычно для этого используются семантически размеченные корпуса). Также применяются комбинации этих методов [17]. Автор работы [18] классифицирует методы разрешения лексической многозначности по типу используемых внешних источников информации:

- структурированные источники данных (машиночитаемые словари, тезаурусы, онтологии);
- неструктурированные источники данных в виде корпусов текстов делятся на:
 - неразмеченные корпуса;
 - синтаксически и/или семантически размеченные корпуса.

Одним из эффективных подходов к решению задачи соотнесения частей текста с узлами онтологии является использование правил, учитывающих контекст, в котором употреблена синтаксема, значение которой требуется определить, и информацию из онтологий — структурированных источников информации.

Будем предполагать, что этапы (s1)–(s6) уже выполнены и мы работаем с набором синтаксем, каждой из которых поставлено в соответствие множество элементов онтологии. Тогда задача разрешения лексической многозначности сводится к тому, чтобы из каждого соответствующего отдельной синтаксеме множества элементов онтологии выбрать один единственный, наилучшим образом отражающий лексическое значение рассматриваемой синтаксемы. Например, для предложения «В лесу у моря стоит замок», задача разрешения лексической многозначности сведется к выбору единственного верного значения из множества элементов RuTez (см. таблицу 1).

Таблица 1. Синтаксемы и соответствующие узлы онтологии RuTez

Название синтаксемы	Узлы онтологии, из которых необходимо выбрать один соответствующий синтаксеме	Узел онтологии, соответствующий лексическому значению синтаксемы
лесу	Рыболовная леска	Лесной массив
	Деловая древесина	
	Лесной массив	
	Лес (множество ч.-нибудь поднятого)	
море	Водный объект	Водный объект
	Море (большое количество)	
	Морить (травить отравой)	
	Морить (мучить, изнурять)	
	Морение древесины	
стоит	Стоять, быть без движения	Находиться, пребывать
	Стоять (бездействовать)	
	Находиться, пребывать	
	Стоять (сохраняться, не портиться)	
	Стоять в вертикальном положении	
	Подобать, надлежать, следовать	
	Стоить, иметь цену	
замок	Замок для запираия	Средневековый замок
	Средневековый замок	

При составлении правил, определяющих соответствие синтаксемы текста элементу онтологии, необходимо учитывать:

- Контекст синтаксемы (ближайший к синтаксеме текст имеет наибольшее значение: наиболее «важным» для анализа является текст предложения, в котором употребляется синтаксема, затем следует текст абзаца, содержащего это предложение, далее – раздел, содержащий упомянутое предложение, затем – раздел более высокого уровня (например, глава или параграф) и т. д., заканчивая всем анализируемым текстом).
- Семантическую близость синтаксемы, соотнесенную в процессе анализа с конкретным элементом онтологии, и синтаксем из контекста, учитывая «близость» контекста к анализируемой синтаксеме; для определения семантической близости можно использовать не только онтологию, но и ассоциативные словари.
- Тематику текста. Для определения тематики текста возможно либо попросить пользователя самого определить ее (например, предложив выбрать из списка), либо определить ее автоматически: в настоящее время существует множество алгоритмов для автоматической рубрикации текстов.

4. Примеры работы системы

4.1. Пример 1

Ниже приведен пример SPARQL-запроса, который формируется программно-реализованной вопросно-ответной системой для следующего вопроса «Какие существуют виды спорта?»:

```
SELECT DISTINCT ?x WHERE { ?sub0 itfriu:normalForm «спорт» .
  ?sub0 owl:sameAs ?samesub0 . ?x rdfs:subClassOf ?samesub0 . }
```

Приведенному SPARQL-запросу, адресованный онтологии РуТез, представленной в формате RDF, будет соответствовать ответ, состоящий из 133 элементов онтологии РуТез. Далее перечислены первые 10 из них: бег на длинную дистанцию, скелетон, прыжки в высоту, бобслейный спорт, баскетбол, легкоатлетический марафон, стендовая стрельба, горнолыжный супергигант, метание молота, хафпайп.

4.2. Пример 2

Исходный текст: Крупный, зеленый, добрый попугай съел кашу и яблоко, а воробей съел грушу

Найденные семантические отношения:

- ЧТО (съел#Г,изъяв,Г,мр,нс,прш,съесть,ед; кашу#С,С,но,каша,жр,вн,ед)
- ПРИЗНАК (попугай#С,С,попугай,мр,им,од,ед;

Крупный#П,но,крупный,мр,им,вн,П,ед,полн)
- СПИСОК (зеленый#П,но,мр,им,вн,зеленый,П,ед,полн;

добрый#П,но,мр,им,добрый,вн,П,ед,полн, #СИМВОЛ,-,СИМВОЛ)
- ДЕЙСТВИЕ (съел#Г,изъяв,Г,мр,нс,прш,съесть,ед;

воробей#С,С,мр,им,од,воробей,ед)
- ПРИЗНАК (попугай#С,С,попугай,мр,им,од,ед;

добрый#П,но,мр,им,добрый,вн,П,ед,полн)
- СПИСОК (кашу#С,С,но,каша,жр,вн,ед; яблоко#С,С,но,им,яблоко,вн,ср,ед,

и#СОЮЗ,СОЮЗ,и)
- ЧТО (съел#Г,изъяв,Г,мр,нс,прш,съесть,ед; грушу#С,С,но,груша,жр,вн,ед)
- ПРИЗНАК (попугай#С,С,попугай,мр,им,од,ед;

зеленый#П,но,мр,им,вн,зеленый,П,ед,полн)
- СПИСОК (Крупный#П,но,крупный,мр,им,вн,П,ед,полн;

зеленый#П,но,мр,им,вн,зеленый,П,ед,полн, #СИМВОЛ,-,СИМВОЛ)
- ЧТО (съел#Г,изъяв,Г,мр,нс,прш,съесть,ед; яблоко#С,С,но,им,яблоко,вн,ср,ед)
- ДЕЙСТВИЕ (съел#Г,изъяв,Г,мр,нс,прш,съесть,ед; попугай#С,С,попугай,мр,им,од,ед)

Заданный вопрос: Какой фрукт съела большая птица?

Ответ: Яблоко

Сгенерированный SPARQL запрос:

```
SELECT DISTINCT ?x ?e WHERE { ?a itfriu:normalForm "фрукт" . ?a owl:sameAs ?b . ?x
  rdfs:subClassOf ?b . ?x itfriu:ns "NEW" . OPTIONAL { ?x itfriu:ПРИЗНАК ?e } . ?x
  itfriu:ЧТО_ИНВ ?x1 . ?x1 rdfs:subClassOf ?x1b . ?x1b owl:sameAs ?x1S . ?x1S itfriu:normalForm
  "съесть" . ?x1 itfriu:ДЕЙСТВИЕ ?x2 . ?x2 rdfs:subClassOf ?x2b . ?x2b owl:sameAs ?x2S . ?x2S
  itfriu:normalForm "птица" . ?x2 itfriu:ПРИЗНАК ?x3 . ?x3 rdfs:subClassOf ?x3b . ?x3b
  owl:sameAs ?x3S . ?x3S itfriu:normalForm "большой" . }
```

4.3. Пример 3

Исходный текст: Большое и красное яблоко

Найденные семантические отношения:

- ПРИЗНАК (яблоко#С,С,но,им,яблоко,вн,ср,ед;
красное#П,красный,им,вн,П,ед,полн,ср)
- СПИСОК (Большое#П,им,большой,вн,П,ед,полн,ср;
красное#П,красный,им,вн,П,ед,полн,ср, и#СОЮЗ,СОЮЗ,и)
- ПРИЗНАК (яблоко#С,С,но,им,яблоко,вн,ср,ед;
Большое#П,им,большой,вн,П,ед,полн,ср)

Заданный вопрос: Какого размера яблоко ?

Ответ: большое

Сгенерированный SPARQL запрос:

```
SELECT DISTINCT ?x ?x1 WHERE { ?a itfru:normalForm "яблоко" . ?a owl:sameAs ?b . ?x
rdfs:subClassOf ?b . ?x itfru:ns "NEW" . OPTIONAL { ?x itfru:ПРИЗНАК ?e } . ?x
itfru:ПРИЗНАК ?x1 . ?x1 rdfs:subClassOf ?x1b . ?x1b owl:sameAs ?x1S . ?x1S itfru:normalForm
"размер" . }
```

4.4. Пример 4

Исходный текст: Слоны обожают бананы, яблоки, морковь, свеклу

Найденные семантические отношения:

- СПИСОК (морковь#С,С,но,им,жр,вн,морковь,ед; свеклу#С,С,но,жр,вн,свекла,ед,
#СИМВОЛ,-,СИМВОЛ)
- СПИСОК (бананы#С,С,но,мн,банан,мр,им,вн; яблоки#С,С,но,мн,им,яблоко,вн,ср,
#СИМВОЛ,-,СИМВОЛ)
- СПИСОК (яблоки#С,С,но,мн,им,яблоко,вн,ср; морковь#С,С,но,им,жр,вн,морковь,ед,
#СИМВОЛ,-,СИМВОЛ)
- ДЕЙСТВИЕ (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
бананы#С,С,но,мн,банан,мр,им,вн)
- ЧТО (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать; свеклу#С,С,но,жр,вн,свекла,ед)
- ЧТО (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
бананы#С,С,но,мн,банан,мр,им,вн)
- ДЕЙСТВИЕ (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
яблоки#С,С,но,мн,им,яблоко,вн,ср)
- ЧТО (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
морковь#С,С,но,им,жр,вн,морковь,ед)
- ДЕЙСТВИЕ (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
Слоны#С,С,мн,мр,им,слон,од)
- ДЕЙСТВИЕ (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
свеклу#С,С,но,жр,вн,свекла,ед)
- ДЕЙСТВИЕ (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
морковь#С,С,но,им,жр,вн,морковь,ед)
- ЧТО (обожают#Г,изъяв,св,мн,буд,Г,пе,3л,обожать;
яблоки#С,С,но,мн,им,яблоко,вн,ср)

Заданный вопрос: Какие фрукты любят животные?

Ответ: банан, яблоко

Сгенерированный SPARQL запрос:

```
SELECT DISTINCT ?x ?e WHERE { ?a itfru:normalForm "фрукт" . ?a owl:sameAs ?b . ?x
rdfs:subClassOf ?b . ?x itfru:ns "NEW" . OPTIONAL { ?x itfru:ПРИЗНАК ?e } . ?x
itfru:ЧТО_ИНВ ?x1 . ?x1 rdfs:subClassOf ?x1b . ?x1b owl:sameAs ?x1S . ?x1S itfru:normalForm
"любить" . ?x itfru:ДЕЙСТВИЕ_ИНВ ?x1 . ?x1 rdfs:subClassOf ?x1b . ?x1b owl:sameAs ?x1S .
?x1S itfru:normalForm "любить" . ?x1 itfru:ДЕЙСТВИЕ ?x2 . ?x2 rdfs:subClassOf ?x2b . ?x2b
owl:sameAs ?x2S . ?x2S itfru:normalForm "животное" . }
```

5. Заключение

В настоящее время вопросно-ответная система, архитектура которой описывается в данной работе, находится в стадии разработки. В будущем планируется провести ряд работ, направленных на улучшение качества работы всех основных модулей системы (морфологический анализ, выделение границ именованных сущностей, соотнесение именованных сущностей с узлами онтологии, поиск семантических зависимостей, поиск ответа на вопрос пользователя). Также планируется провести полноценное тестирование системы, на больших объемах данных — например, на множестве статей с известных новостных сайтов. Подобный выбор тестовых данных весьма распространен среди разработчиков систем, нацеленных на извлечение информации из текстов. Например, в работе [19] авторы в качестве анализируемых корпусов текстов предлагают рассматривать статьи, опубликованные на известных новостных сайтах CNN и Daily Mail — этот набор данных стал стандартным для задач понимания текстов.

Литература

- [1] Мочалова А.В. Семантический анализатор русскоязычного текста для вопросно-ответной системы: дис. ... канд. техн. наук: 05.13.18. Петрозаводск, 2017. 128 с.
- [2] Мозговой М.В. Простая вопросно-ответная система на основе семантического анализатора русского языка // Вестник СПб университета. 2005. Сер. 10. Вып. 1. С. 116-122.
- [3] Кулешов С.В. Вариант архитектуры субпоисковой системы для реализации функции аналитического мониторинга / С.В. Кулешов, С.Н. Михайлов // Труды СПИИРАН. 2013. № 8(31). С. 247-254.
- [4] Захаров В.П., Мочалова А.В., Мочалов В.А. Вопросно-ответные системы. Некоторые проблемы автоматической обработки текста. Петрозаводск: ПИН, 2015.
- [5] Лингвистическая онтология «Тезаурус РуТез» // URL: <http://www.labinform.ru/pub/ruthes/index.htm> (дата обращения 17.02.2018).
- [6] Kuznetsov V.A. Ontological-semantic text analysis and the question answering system using data from ontology / Kuznetsov V.A., Mochalov V.A., Mochalova A.V. // ICACT Transactions on Advanced Communications Technology (TACT). 2015. Vol. 4, Issue 4. P. 651-658.
- [7] Mochalova A.V., Mochalov V.A. Mathematical model of an ontological-semantic analyzer using basic ontological-semantic patterns // Lecture Notes in Artificial Intelligence, Proceedings of 15th Mexican International Conference on Artificial Intelligence. 2016. P. 53-66.
- [8] Экспертная система Drools // URL: <https://www.drools.org> (дата обращения 17.02.2018).
- [9] Apache Jena // URL: <https://jena.apache.org/> (дата обращения 17.02.2018).
- [10] SPARQL Query Language for RDF // URL: <https://www.w3.org/TR/rdf-sparql-query/> (дата обращения 17.02.2018).
- [11] Богуславский И.М. и др. Семантический анализ и ответы на вопросы: система в стадии разработки / И.М. Богуславский, В.Г. Диконов, Л.Л. Иомдин, А.В. Лазурский и др. // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог» (Москва, 27–30 мая 2015 г.). Вып. 14 (21): В 2 т. М.: Изд-во РГГУ, 2015. Т. 1. С. 62 – 79.
- [12] Автоматическая обработка текста // URL: <http://www.aot.ru>. (дата обращения 10.03.2018).
- [13] Урюпина О. Автоматическое разбиение текста на предложения для русского языка // Компьютерная лингвистика и интеллектуальные технологии: По материалам ежегодной Международной конференции «Диалог». Вып. 7 (14). М.: РГГУ, 2008.

- [14]Золотова Г.А. Синтаксический словарь: Репертуар элементарных единиц русского синтаксиса. М.: Наука, 1988.
- [15]Зализняк А.А. Грамматический словарь русского языка. Словоизменение. М.: Русский язык, 1977.
- [16]Каушинис Т.В. и др. Обзор методов и алгоритмов разрешения лексической многозначности: Введение / Каушинис Т.В., Кириллов А.Н., Коржицкий Н.И., Крижановский А.А., Пилинович А.В., Сихонина И.А., Спиркова А.М., Старкова В.Г., Степкина Т.В., Ткач С.С., Чиркова Ю.В., Чухарев А.Л., Шорец Д.С., Янкевич Д.Ю., Ярышкина Е.А. // Труды КарНЦ РАН. № 10. Сер. Математическое моделирование и информационные технологии. 2015. С. 69-98
- [17]Лукашевич Н. В. Тезаурусы в задачах информационного поиска. М.: МГУ, 2011. 495 с.
- [18]Navigli R. Word sense disambiguation: A survey. // ACM Computing Surveys (CSUR). 2009. Vol. 41, № 2.
- [19]Hermann K. M. and etc. Teaching machines to read and comprehend / Hermann K. M., Kocisky T., Grefenstette E., Espeholt L., Kay W., Suleyman M., and Blunsom P. // Advances in Neural Information Processing Systems. 2015. P. 1684–1692.

Software Implementation of Question-Answering System that Uses Ontology Data on the Basis of Apache Jena Framework

A.V. Mochalova, V.A. Mochalov

Institution Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS

The development of question-answer systems that allow users to answer questions asked in natural language on machine-readable texts is very urgent task.

The paper describes a question-answer system that uses data from RuTez ontology. The paper describes the stages of solving the problem of mapping parts of text with ontology nodes. There are 6 stages in total: preliminary processing of the text; definition of the boundaries of sentences; allocation of syntaxemes boundaries; determination of possible lemmas variants for all allocated syntaxemes; search in the ontology for elements corresponding to initial forms of the syntaxemes; selection among the ontology elements corresponding to the syntaxemes.

The description of the question-answer system architecture based on Apache Jena, Drools expert system and semantic analyzer developed by authors is provided. End-to-end examples of the system operation are given.

Keywords: question-answer systems, ontologies, automatic text processing, Drools, SPARQL, Apache Jena