УДК: 004.932, 517.5

1.6. ИЗВЛЕЧЕНИЕ ДАННЫХ ИЗ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ПОМОЩЬЮ СПЕЦИАЛЬНОГО ЯЗЫКА ЗАПРОСОВ И ЛИНГВИСТИЧЕСКОЙ ОНТОЛОГИИ

Ночевной Д.С.[1]
[1]Национальный исследовательский университет ИТМО

В данной статье рассматривается применение специального языка запросов для извлечения словоформ, полученных из слабоструктурированных источников, рассматриваются основные термины в области семантических сетей. В исследовании приводится обзор существующих программных продуктов для синтаксического анализа предложений в текстах на русском языке. Также представлено описание программной системы, позволяющей преобразовывать тексты на естественном языке в унифицированный формат и выполнять извлечение данных с помощью специального языка запросов и существующей лингвистической онтологии. Новизну исследования составляет использование «смысловых единиц» из лингвистической онтологии, обеспечивающих более высокое качество (пертинентность) извлечения требуемой информации. В качестве итога приведены результаты оценки точности и полноты разработанного алгоритма для извлечения данных.

Введение

Формализовать знания в определенной предметной области и систематизировать их для быстрого доступа всегда являлось важной и актуальной задачей. Во многих научных областях такие системы знаний принято использовать для того, чтобы помочь пользователям легко и оперативно получать требуемую информацию [Письмак, 2016].

Для построения современных онтологий всегда актуальным является извлечение семантически связанных слов из текста на естественном языке, поскольку текст наряду со словарями является одним из важнейших источников информации. Однако общим недостатком многих онтологий является отсутствие специализированных терминов, специфичных для данной предметной области.

Чаще всего онтологии дополняются специализированными терминами путем ручного занесения узлов и семантических связей. Но этот подход по определению является малоэффективным и весьма затруднительным, ведь создание онтологии достаточно большого объема и ее постоянная поддержка в актуальном состоянии требует больших трудозатрат. Очевидно, что ручной сбор данных для онтологий – это утомительная и громоздкая задача из-за очень большого объема этих данных и количества связей между ними. Более того, подобная деятельность требует глубокого знания предметной области. Вследствие этого факта, во множестве случаев полученный результат может не иметь требуемой точности и полноты.

Поэтому и существует проблема дополнения онтологий новыми узлами и связями, а значит, появляется также и проблема извлечения информации из слабоструктурированных источников.

Слабоструктурированные данные возникают, когда источник не налагает жесткую структуру и когда данные объединяются из нескольких гетерогенных источников [Карташов, 2018]. В отличие от традиционных хорошо структурированных данных, схема которых известна заранее, слабоструктурированные данные не имеют фиксированную схему. Они характеризуются наличием гибкой структуры, которая определяет их неоднородное содержание. Структура слабоструктурированных документов часто подразумевается, а не является жесткой или полной, как в случае традиционных баз данных.

Целью работы стало повышение качества (пертинентности) извлекаемой информации из слабоструктурированных источников. А в рамках поставленной цели были сформулированы задачи:

- анализ существующих программных модулей, позволяющих провести синтаксический разбор предложений для текстов на русском языке;
- разработка программного модуля для преобразования текстов на естественном языке в структурированный унифицированный формат;
- добавление в язык запросов поддержки использования семантической сети для извлечения данных с помощью «смысловых блоков» («смысловых единиц»).

Онтологии и семантические сети

В статье «Применение семантической сети для хранения слабоструктурированных данных» [Клименков, 2020] рассматривается структура существующей лингвистической онтологии. В общем смысле семантическая сеть состоит из множества смысловых понятий или концептов, связанных между собой семантическими отношениями или связями. Концепты и связи могут иметь свойства или атрибуты, характеризующие их. Концепты, связи и атрибуты образуют структурный компонент семантической сети. Кроме этого, концепты, связи и атрибуты могут иметь экземпляры, образующие информационный компонент семантической сети.

В предыдущем исследовании [Ночевной, 2022] приводится более подробное описание семантических сетей и их элементов, в том числе различные типы семантических отношений. Также в нём рас-

сматривается первая версия специального языка запросов и примеры выражений, содержащих все возможные типы блоков.

В рамках данного исследования используются предыдущие разработки, но теперь специальный язык запросов получил своё развитие, позволив сформировать целостную систему для осуществления поиска данных даже в слабоструктурированных источниках.

Обзор существующих решений для синтаксического разбора текста

В предыдущей работе [Ночевной, 2022] в качестве исходных данных был взят национальный корпус текстов русского языка, или, выражаясь более точно, его синтаксически размеченный подкорпус «СинТагРус».

Однако «СинТагРус» является относительно небольшим, и было решено поддержать поиск даже по слабоструктурированным файлам с текстом на естественном языке. Для преобразования текста на естественном языке в унифицированный формат, используемый для поиска в тексте информации с помощью специальных регулярных выражений, было предложено разработать конвертер на языке Java.

К счастью, существует немало готовых решений (программных модулей), позволяющих делать такие преобразования[1] [Мухамедиев, 2018]. Поэтому не было необходимости разрабатывать конвертер с нуля, а нужно было лишь выбрать наиболее подходящий по критериям и дополнить его работу необходимыми шагами. Рассмотрим существующие программные модули, позволяющие провести синтаксический разбор предложений для русского языка.

В числе критериев для выбора подходящего инструмента можно выделить применимость к текстам на русском языке, поскольку исходная онтология была сформирована именно на русском языке и в дальнейшем онтология будет полезна для получения дополнительных данных и построения более эффективных поисковых запросов. По данному критерию не подошли «MSTParser», «NLTK», «MBSP», «Pattern», «The Stanford Parser», «ZPar», «mate-tools».

Кроме того, немаловажным условием было то, что программный модуль должен иметь некоммерческую лицензию или, по крайней мере, лицензию, позволяющую использовать данный программный модуль для исследовательских и учебных целей. По данному критерию были исключены модули «Solarix», «ABBYY Compreno», «AskNet», «DictaScope» и «Синтактико-Семантический Анализ Русского Языка».

Наконец, необходимо было обеспечить работу выбранного решения как на базе операционных систем Windows, так и на базе операционных систем Linux. При этом предполагалось, что в данном случае также подойдёт библиотека на языке Java или какой-либо программный интерфейс приложения (Application Programming Interface), позволяющий работать через один из популярных протоколов. При этом должен быть возможен локальный запуск, так как обращение к веб-сервису в интернете привело бы к значительным задержкам в работе. Критерий кроссплатформенности является немаловажным, поэтому он также заставил нас отсеять программные модули «ISPRAS API Texterra», доступный в виде веб-сервиса в интернете, и «ЭТАП-3», доступный только для операционных систем семейства Windows, а также «Link Grammar Parser», предоставляющий веб-сервис для анализа русского языка.

А теперь рассмотрим более подробно оставшиеся программные модули из списка: «AOT», «MaltParser», «AGFL» и «zamgi».

Проект «AOT»[2] является достаточно известным, однако бинарные дистрибутивы программ не поддерживаются с 2021 года. Можно было бы скомпилировать исходный код на языке C/C++ вручную под конкретную платформу, но с учетом трудозатрат было решено использовать другой программный модуль. Проект «AOT» – это российское ПО для решения задач обработки естественно-языковых текстов [Мухамедиев, 2018].

Проект «MaltParser» [Nivre, 2007] не обновляется уже длительное время, тем не менее, модели для разных языков поддерживаются энтузиастами, а для скачивания доступно всё необходимое: бинарные файлы, докер-образ и даже Java-библиотека. MaltParser[3] — инструмент для работы с деревьями зависимостей. Он позволяет построить модель по размеченному корпусу и строить деревья для новых данных, основываясь на ней. Реализует несколько алгоритмов построения деревьев. Сергей Шаров подготовил всё необходимое для обработки русского языка от голого текста до получения дерева зависимостей с помощью MaltParser, попутно описав основные проблемы в работе с русским языком.

Проект «AGFL» (Affix Grammars over a Finite Lattice) [Koster, 2002] имеет свою страницу в интернете, однако не удалось загрузить какие-либо бинарные файлы для использования или исходный код. Система AGFL (Affix Grammar for Finite Lattice) – свободно распространяемое программное обеспечение для решения задач автоматической обработки текстов на естественном языке, использующее

[1] Обработка текста – NLPub URL: https://nlpub.ru/Обработка_текста#Синтаксический_анализ (дата обращения 25.04.2023).
[2] AOT :: Главная // Автоматическая обработка текста URL: http://aot.ru (дата обращения 25.04.2023).
[3] MaltParser — NLPub URL: https://nlpub.ru/MaltParser (дата обращения 25.04.2023).

формализм AGFL. Система была разработана на отделении Компьютерных исследований Университета Неймеген (Нидерланды) под руководством К. Костера [Азарова, 2003].

Проект «Zamgi»[4] также является достаточно интересным, поддерживающим русский язык, а также кроссплатформенным, однако реализован на языке C#, поэтому не подходит, ведь исходный код, отвечающий за преобразование текстов в унифицированный формат, реализован на языке Java. NER-ru – сервис, созданный человеком или группой под псевдонимом Zamgi на сайте GitHub [Мухамедиев, 2018].

Поэтому итоговый выбор был сделан в пользу инструмента «MaltParser», поскольку его использование возможно локально, без какого-либо подключения к внешним сервисам в интернете. Кроме того, достаточно удобным в данном случае является наличие библиотеки на языке Java. Нужно лишь загрузить обученные модели и разместить их в правильном месте на диске, чтобы можно было использовать токенизацию и синтаксический анализ непосредственно в исходном коде конвертера текстовых документов в требуемый унифицированный формат.

Разработка модуля для преобразования текстов и извлечения данных

В рамках данного исследования был разработан конвертер на языке Java для преобразования текста на естественном языке в унифицированный формат, используемый для поиска в тексте информации с помощью специальных регулярных выражений.

Собственно, на рисунке 1 видно, что модуль программы «Конвертер текста» в качестве основного элемента включает в себя модуль «MaltParser». В данном случае было решено подключить модуль «MaltParser» в программу в виде библиотеки на языке Java. Эта библиотека является так называемой «программной обёрткой», которая инкапсулирует в себе работу с обученной моделью для преобразования передаваемых предложений в токены и их классификации. То есть для корректной работы данного модуля необходимо иметь доступ к токенизатору и синтаксическому анализатору.

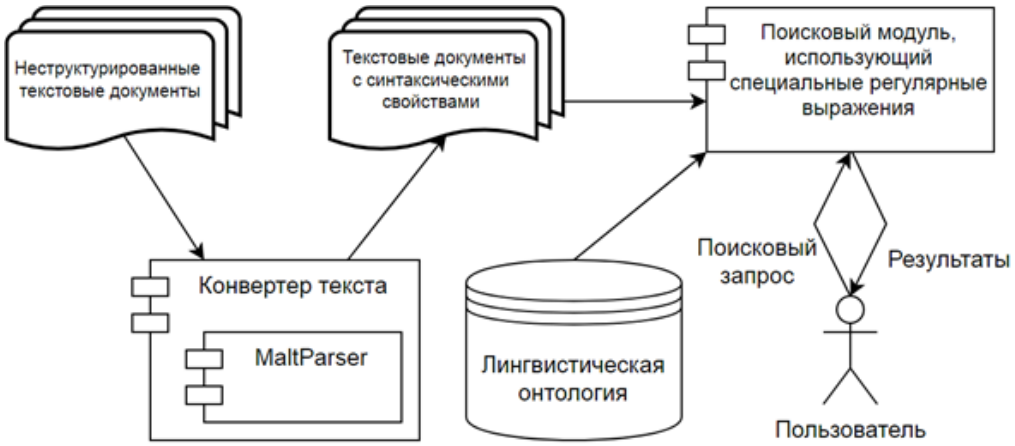

Рис. 1. Преобразование текста на естественном языке в формат, удобный для поиска

Рассмотрим также один пример запроса с использованием специальных регулярных выражений, отличающийся наличием обращений к лингвистической онтологии. Подобные обращения могут потребоваться, когда мы ищем данные с помощью так называемых «смысловых единиц», или «сенсов».

На рисунке 2 представлен поисковый запрос с использованием двух «смысловых единиц» – «здание_1» и «жилище_1». Соответственно, каждая из этих смысловых единиц может быть представлена в тексте в виде множества словоформ, означающих заданный смысл. Например, «здание_1» в тексте может быть выражено лексемами «строение», «дом», «здание», «постройка» и другими. При этом напомним, что лексема – это множество словоформ одного и того же слова, но в различных формах. Например, это может быть существительное в разных падежах.

Результатом выполнения запроса на рисунке 2 могут быть следующие предложения, при этом жирным шрифтом выделены ключевые словоформы, соответствующие заданным «смысловым единицам».

- В этом **доме** есть четыре жилые **комнаты**, а также кухня, гостиная и другие.
- Это **здание** имеет несколько складских **помещений**, используемых в основном для хранения электронной техники.

[4] GitHub – zamgi/lingvo--Syntax-ru: Определение синтаксических ролей слов в предложении в тексте на русском языке URL: https://github.com/zamgi/lingvo--Syntax-ru (дата обращения 25.04.2023).

Рис. 2. Пример поискового запроса с использованием «смысловых единиц» в составе специального регулярного выражения

Практическое использование языка запросов и оценка точности

После преобразования текстов с помощью экспертов-лингвистов стало возможным определить корректные и некорректные результаты, получаемые с помощью «смысловых блоков», входящих в состав специальных регулярных выражений.

Для проверки точности и полноты разработанного поискового механизма с использованием «смысловых блоков, или «смысловых единиц, было предложено использовать F-меру.

Рассмотрим пример запроса для поиска всех словоформ, которыми может быть выражен смысловой блок «{здание_1}», сопровождаемый одним прилагательным слева. Запрос будет соответствовать любой словоформе в любом числе и падеже, которая несёт в себе смысл «здание_1» (постройка). Примерами таких словоформ могут быть «здание» или «дом».

- python3 regexp_patterns.py '[A]{здание_1}' file.json

Пример результатов работы команды:

- многоквартирный **дом**
- прочный **здание**
- кирпичный **здание**
- загородный **дом**
- жилой **дом**

Рассмотрим также пример одного из запросов, которые использовались в экспериментах для определения точности и полноты поиска с использованием сразу двух смысловых блоков. Для одного из таких запросов было использовано выражение «.*{здание_1}.*{площадь_1}.*». В данном случае подразумевается поиск всех предложений, в состав которых входят словоформы, выражающие два заданных «смысловых блока», именно в указанном порядке, причём перед ними, между ними и после них может быть любое количество других словоформ, что соответствует элементам «.*».

- python3 regexp_patterns.py '.*{здание_1}.*{площадь_1}.*' file.json

Пример результатов работы команды:

- у он быть **дом** в центр город который располагаться на **площадь** 200 квадратный метр
- прочный **здание площадь** более 100 квадратных метров быть использовать как склад

Для проверки работы разработанного «конвертера текста» и языка запросов в реальных условиях были выбраны 10 тестовых запросов с использованием «смысловых единиц» и около 600 преобразованных текстов в унифицированном формате. Среди полученных данных оказалось 143 результата, которые соответствовали запросам пользователя. Остальные 58 результатов были некорректными. На основе полученных данных была составлена матрица ошибок, приведённая в таблице 1. По матрице ошибок были вычислены точность, полнота и так называемая F-мера. Точность (precision) составила: TP / (TP + FP) = 141 / (141 + 56) = 0.716, полнота (recall) оказалась равной: TP / (TP + FN) = 141 / (141 + 26) = 0.844, а F-мера в свою очередь при коэффициенте β = 1 будет иметь значение: 2 * precision * recall / (precision + recall) = 2 * 0.716 * 0.844 / (0.716 + 0.844) = 0.774.

Следует в первую очередь обратить внимание на некоторые некорректные результаты поиска. Во-первых, это те результаты, которые относятся к так называемой группе «False Positive», то есть определённые как корректные результаты, но на самом деле не удовлетворяющие поисковому запросу пользователя. Есть несколько примеров таких результатов. Давайте перечислим основные типы.

Таблица 1. Матрица ошибок, полученная после тестирования «смысловых единиц» в составе специальных регулярных выражений

	True	False
Positive	141	56
Negative	-	26

Омонимия. Нередко какая-либо словоформа, выражающая искомый «смысловой блок», может относиться также и к другим «смысловым блокам». Например, словоформа «дорога» может быть как существительным («эта дорога ведёт прямо к центру города»), так и кратким прилагательным («эта книга очень дорога мне»). Данный недостаток можно частично устранить, добавив в «смысловые блоки» также информацию о требуемых частях речи и их признаках.

Словоформы, относящиеся к другой части предложения. При преобразовании текстов на естественном языке в унифицированный формат, к сожалению, у каждого предложения утрачиваются связи между конкретными членами предложения, и поиск с использованием «специальных выражений» не позволяет отделять разные части сложных предложений. Например, в предложении «около дороги был дом, который стоял на участке площадью шесть соток» есть словоформы «дом» и «площадью», однако они не связаны по смыслу напрямую. Влияние этой особенности можно частично устранить, указывая максимальное расстояние между искомыми словами в предложении. Другим способом решения данной проблемы является доработка модуля «конвертер текста» таким образом, чтобы поддержать структуру сложных предложений, а также связи между членами простых предложений.

Неполнота имеющейся лингвистической онтологии. Не исключено, что для какого-либо «смыслового блока» в лингвистической онтологии представлены не все возможные словоформы. Также могут просто отсутствовать необходимые семантические связи, которые должны были быть между узлами онтологии.

Во-вторых, существует также категория результатов, которые относятся к группе «False Negative», то есть это те результаты, которые должны были быть получены в рамках ответа на запрос, но не были найдены программой. В нашем случае элементы данной категории встречаются реже, чем элементы категории «False Positive», но их источники также необходимо рассмотреть и постараться устранить в будущем. Перечислим основные проблемы данной категории.

Использование местоимений вместо искомых словоформ. Все тексты на естественных языках подвержены этому. Люди склонны упрощать тексты, как для избегания повторений, так и для упрощения чтения этих текстов. Например, «Эта книга есть и в моей домашней коллекции. Она очень дорога мне». Чтобы алгоритм мог обрабатывать такие сложные случаи, необходимо поддерживать некоторый контекст, содержащий факты из уже «прочитанных» предложений. В таком случае программа будет понимать, что подразумевается под тем или иным местоимением.

Пропуск некоторых слов, без которых смысл всё равно остаётся понятным для человека. Иногда к этому прибегают также и авторы художественных произведений для придания особого колорита и эстетической выразительности своим произведениям. Например, «Квартира была [площадью] 80 квадратных метров». В данном случае любой читатель понимает смысл этого предложения, несмотря на то, что слово «площадью» опущено. А вот поисковый алгоритм пропустил бы данное предложение, так как одна из обязательных частей поискового запроса отсутствует. Устранение данного недостатка также не является тривиальной задачей, так как для «понимания» полного смысла предложения программой необходимо вводить некоторый контекст.

Заключение

В рамках исследования было рассмотрено решение задачи по добавлению в язык поддержки использования семантической сети для извлечения данных с помощью смысловых единиц. В результате исследования был разработан метод поиска словоформ в структурированных текстах с использованием смысловых единиц из лингвистической онтологии, обеспечивающий более высокую пертинентность извлечения требуемой информации.

Полученный результат подтвердил гипотезу о том, что использование «смысловых единиц» в специальных регулярных выражениях может повлиять на точность и полноту получаемых данных и повысить пертинентность поиска, если учитывать данные из существующей лингвистической онтологии. Следует отметить, что для достижения большей точности нужно использовать не все существующие лексемы и словоформы для заданных «смысловых единиц», так как в ряде случаев в результатах поиска появляется избыточность.

Таким образом, в процессе исследования был проведен автоматизированный поиск данных в тестовом корпусе текстов. Подтвердилась гипотеза о том, что использование смысловых блоков может повысить пертинентность поиска в текстах на естественном языке. Полученная точность составила 0.716, полнота оказалась равной 0.844. Значение F-меры разработанного модуля для извлечения данных составило около 0.774. Полученные результаты показывают практическую значимость предложенного метода и возможность его успешного применения для извлечения данных из текста.

Литература

1. Азарова И. В. Морфологическая разметка текстов на русском языке с использованием формальной грамматики AGFL //Компьютерная лингвистика и интеллектуальные технологии. Труды Международной конференции. Диалог. – 2003. – С. 51-55.
2. Карташов О.О., Бутакова М.А., Чернов А.В., Костюков А.В., Жарков Ю.И. Средства представления знаний и извлечения данных для интеллектуального анализа ситуаций // Инженерный вестник Дона. 2018. №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5421

3. Клименков С.В., Николаев В.В., Харитонова А.Е., Гаврилов А.В., Письмак А.Е., Покид А.В. Применение семантической сети для хранения слабоструктурированных данных // Инженерный вестник Дона. 2020. №2. URL: ivdon.ru/ru/magazine/archive/N2y2020/6339
4. Мухамедиев Равиль Ильгизович, Сымагулов Адилхан, Кучин Ян Игоревич, Абдуллаева Сабина, Абдолдина Фарида Наурузбаевна Облачные сервисы для обработки текстов на естественном языке // Современные информационные технологии и ИТ-образование. 2018. №4. URL: https://cyberleninka.ru/article/n/oblachnye-servisy-dlya-obrabotki-tekstov-na-estestvennom-yazyke (дата обращения: 25.04.2023).
5. Ночевной Д. С., Бессмертный И. А., Клименков С. В. Специальные выражения для поиска в структурированном тексте с использованием грамматических свойств // Цифровая экономика [Электронный ресурс]. – 2022. – URL: http://digital-economy.ru/stati/специальные-выражения-для-поиска-в-структурированном-тексте-с-использованием-грамматических-свойств (дата обращения 25.04.2023).
6. Обработка текста – NLPub [Электронный ресурс]. – 2018. – URL: https://nlpub.ru/Обработка_текста#Синтаксический_анализ (дата обращения 25.04.2023).
7. Письмак А.Е., Харитонова А.Е., Цопа Е.А., Клименков С.В. Метод автоматического формирования семантической сети из слабоструктурированных источников // Программные продукты и системы. 2016. №3. С. 74-78.
8. AOT :: Главная // Автоматическая обработка текста [Электронный ресурс]. – 2021. – http://aot.ru (дата обращения 25.04.2023).
9. GitHub – zamgi/lingvo--Syntax-ru: Определение синтаксических ролей слов в предложении в тексте на русском языке [Электронный ресурс]. – 2023. – URL: https://github.com/zamgi/lingvo--Syntax-ru (дата обращения 25.04.2023).
10. Koster C. H. A. et al. The AGFL Grammar Work Lab //USENIX Annual Technical Conference, FREENIX Track. – 2002. – С. 13-18.
11. MaltParser — NLPub [Электронный ресурс]. – 2022. – https://nlpub.ru/MaltParser (дата обращения 25.04.2023).
12. Nivre J. et al. MaltParser: A language-independent system for data-driven dependency parsing //Natural Language Engineering. – 2007. – Т. 13. – №. 2. – С. 95-135.

References in Cyrillics

1. Azarova I. V. Morfologicheskaya razmetka tekstov na russkom yazyke s ispol'zovaniem for-mal'noj grammatiki AGFL //Komp'yuternaya lingvistika i intellektual'nye tekhnologii. Trudy Mezhdunarodnoj konferencii. Dialog. – 2003. – S. 51-55.
2. Kartashov O.O., Butakova M.A., CHernov A.V., Kostyukov A.V., ZHarkov YU.I. Sredstva predstavleniya znanij i izvlecheniya dannyh dlya intellektual'nogo analiza situacij // Inzhenernyj vestnik Dona. 2018. №4. URL: ivdon.ru/ru/magazine/archive/n4y2018/5421
3. Klimenkov S.V., Nikolaev V.V., Haritonova A.E., Gavrilov A.V., Pis'mak A.E., Pokid A.V. Primenenie semanticheskoj seti dlya hraneniya slabostrukturirovannyh dannyh // Inzhener-nyj vestnik Dona. 2020. №2. URL: ivdon.ru/ru/magazine/archive/N2y2020/6339
4. Muhamediev Ravil' Il'gizovich, Symagulov Adilhan, Kuchin YAn Igorevich, Abdullaeva Sabi-na, Abdoldina Farida Nauruzbaevna Oblachnye servisy dlya obrabotki tekstov na este-stvennom yazyke // Sovremennye informacionnye tekhnologii i IT-obrazovanie. 2018. №4. URL: https://cyberleninka.ru/article/n/oblachnye-servisy-dlya-obrabotki-tekstov-na-estestvennom-yazyke (data obrashcheniya: 25.04.2023).
5. Nochevnoj D. S., Bessmertnyj I. A., Klimenkov S. V. Special'nye vyrazheniya dlya poiska v strukturirovannom tekste s ispol'zovaniem grammaticheskih svojstv // Cifrovaya ekonomika [Elektronnyj resurs]. – 2022. – URL: http://digital-economy.ru/stati/special'nye-vyrazheniya-dlya-poiska-v-strukturirovannom-tekste-s-ispol'zovaniem-grammaticheskih-svojstv (data obrashcheniya 25.04.2023).
6. Obrabotka teksta – NLPub [Elektronnyj resurs]. – 2018. – URL: https://nlpub.ru/Obrabotka_teksta#Sintaksicheskij_analiz (data obrashcheniya 25.04.2023).
7. Pismak A.E., Haritonova A.E., Copa E.A., Klimenkov S.V. Metod avtomaticheskogo formiro-vaniya semanticheskoj seti iz slabostrukturirovannyh istochnikov // Programmnye produkty i sistemy. 2016. №3. S. 74-78.
8. AOT :: Glavnaya // Avtomaticheskaya obrabotka teksta [Elektronnyj resurs]. – 2021. – http://aot.ru (data obrashcheniya 25.04.2023).
9. GitHub – zamgi/lingvo--Syntax-ru: Opredelenie sintaksicheskih rolej slov v predlozhenii v tekste na russkom yazyke [Elektronnyj resurs]. – 2023. – URL: https://github.com/zamgi/lingvo--Syntax-ru (data obrashcheniya 25.04.2023).
10. Koster C. H. A. et al. The AGFL Grammar Work Lab //USENIX Annual Technical Conference, FREENIX Track. – 2002. – S. 13-18.
11. MaltParser — NLPub [Elektronnyj resurs]. – 2022. – https://nlpub.ru/MaltParser (data obrashcheniya 25.04.2023).

12. Nivre J. et al. MaltParser: A language-independent system for data-driven dependency parsing //Natural Language Engineering. – 2007. – Т. 13. – №. 2. – S. 95–135.

Ключевые слова
Семантическая сеть, онтология, семантическая связь, регулярные выражения, язык запросов

Ночевной Дмитрий Сергеевич,
аспирант факультета программной инженерии и компьютерной техники НИУ ИТМО,
182506@niuitmo.ru

Keywords
Semantic network, ontology, semantic relation, regular expressions, query language

DOI: DE-2023-03-06
JELclassification – C81 Методология сбора, оценки и организации микроэкономических данных, анализ данных; C82 Методология сбора, оценки и организации макроэкономических данных, анализ данных; C87 Эконометрическое программное обеспечение; D83 Поиск, обучение, информация и знания, взаимодействие, мнение, неосведомленность

Abstract
This article is devoted to a specially designed query language for extracting word forms obtained from semi-structured sources; it contains description of the main terms in the field of semantic networks. The study provides an overview of existing software products for parsing sentences in texts in Russian. A description of the developed software system is also presented. It allows converting natural language texts into a unified format and performing data extraction by using the special query language and the existing linguistic ontology. The novelty of the study is in usage of "semantic blocks" from the linguistic ontology, providing a higher quality (pertinence) of extracting the required information. As a result, accuracy and recall of the developed algorithm for extracting data are presented.