УДК 681.3.06 https://doi.org/10.15407/pp2020.02-03.322

МЕТОД ПОЛУЧЕНИЯ ИНФОРМАЦИИ ИЗ ОНТОЛОГИИ НА ОСНОВЕ АНАЛИЗА ФРАЗЫ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

А.А. Литвин, В.Ю. Величко, В.В. Каверинский

Разработан метод анализа фраз на естественных языках флективного типа (украинский и русский), позволяющий выделить в предложениях основные идеи и группы слов, при помощи которых они излагаются. Сформированные таким образом семантические деревья высказываний, каждое из которых выражает одну конкретную идею, являются удобным исходным материалом для построения запросов к онтологии на языке SPARQL. Метод анализа предложений включает следующую последовательность основных этапов: разбиение на слова, выделение маркерных слов и словосочетаний, определение типа высказывания, выделение именных групп, составление синтаксического графа предложения, построение семантических деревьев высказываний, основанных на имеющихся типах высказываний, подстановка параметров из семантических деревьев высказываний в соответствующие шаблоны SPARQL запросов. Выбор соответствующего шаблона запроса зависит от типа высказывания, выраженного данным семантическим деревом высказывания. Понятия, полученные в качестве ответа на запрос, связываются с соответствующим семантическим деревом высказывания. В случае неполучения информации из онтологии, производится редукция именных групп для выражения более общих понятий и построение запросов с их использованием. Это позволяет всегда получить некоторый ответ, хотя и не столь точный, как при использовании полной именной группы. Использование шаблонов SPARQL запросов требует априорно заданной структуры онтологии, которая также предлагается в данной работе. Такая система применима для ведения диалога с помощью чат-бота, для автоматического получения ответов на вопросы к тексту.

Ключевые слова: онтология, SPARQL, анализ текста, именная группа, синтаксический граф, семантическое дерево высказывания, NLP, NLU.

Розроблено метод аналізу природно-мовних речень для мов флективного типу (українська та російська), який дозволяє виділити в реченні основні висловлені ідеї та групи слів, за допомогою яких вони викладаються. Сформовані таким чином семантичні дерева висловлювань, кожне з яких виражає одну конкретну ідею, є зручним вихідним матеріалом для побудови запитів до онтології мовою SPARQL. Метод аналізу речень включає наступну послідовність основних етапів: розбиття на слова, виділення маркерних слів і словосполучень, визначення типу висловлювання, виділення іменних груп, побудова синтаксичного графа речення, побудова семантичних дерев висловлювань, заснованих на наявних типах висловлювань, підстановка параметрів з семантичних дерев висловлювань до відповідних шаблонів SPARQL запитів. Вибір відповідного шаблону запиту залежить від типу висловлювання, яке виражено семантичним деревом висловлення. Отримані в якості відповіді на запит набори понять зв'язуються з відповідним семантичним деревом висловлювання. У разі неотримання інформації з онтології, проводиться редукція іменних груп для вираження більш загальних понять і побудови запитів з їх використанням. Це дозволяє завжди отримати деяку відповідь, хоч і не настільки точну, як при використанні повної іменної групи. Використання шаблонів SPARQL запитів вимагає апріорно заданої структури онтології, яка також пропонується в даній роботі. Така система може бути застосована для організації діалогу з чат-ботом або для автоматичного отримання відповідей на питання до тексту.

Ключові слова: онтологія, SPARQL, аналіз тексту, іменна група, синтаксичний граф, семантичне дерево висловлення, NLP, NLU.

A method for phrases analyzing in natural languages of inflective type (Ukrainian and Russian) has been developed. The method allows one to outline main expressed ideas and groups of words in the text by which they are stated. The semantic trees of propositions formed in this way, each of which expresses one specific idea, are a convenient source material for constructing queries to the ontology in the SPARQL language. The analysis algorithm is based on the following sequence of basic steps: word tokenize, determining of marker words and phrases, identifying available type of proposition, identifying nouns groups, building a syntactic graph of a sentence, building semantic trees of propositions based on existing types of propositions, substituting parameters from semantic trees of propositions in the corresponding SPARQL query templates. The choice of an appropriate template depends on the type of proposition expressed by a given semantic tree of a proposition. The sets of concepts received as an answer are tied as corresponding answers to the previously defined semantic tree of proposition. In case of non-receipt of information from the ontology, the reduction of noun groups is carried out to express more general concepts and the building queries using them. This allows us to get some answer, although not as accurate as when we use the full noun group. The use of SPARQL query templates requires an a priori known ontology structure, which is also proposed in this paper. Such a system is applicable for dialogue using chat-bots or for automatically receiving answers to questions from the text.

Key words: ontology, SPARQL, text analysis, noun group, syntactic graph, semantic tree of a proposition, NLP, NLU.

Введение

В данной работе рассматривается применение методов анализа естественно-языковых текстов, написанных на языках флективного типа, в частности, украинском и русском. Языковой анализ рассматривается в аспекте выявления в тексте основных посылов и намерений, что необходимо для построения запросов к базе знаний для предоставления релевантной информации, используемой для построения ответов на запрос пользователя. Таким образом, работа посвящена применению методов NLP (natural language processing) и NLU (natural language understanding) для решения задачи организации диалога пользователя с базой знаний на естественном языке. Кроме того, уделяется внимание работе с базами знаний, представленными в онтологической форме.

Центральным компонентом интеллектуальной экспертной системы является база знаний. Такие системы предназначены для поиска решения проблем в некоторой предметной области, основываясь на модели предметной области, созданной пользователем-экспертом. Базы знаний работают совместно с системами поиска, извлечения и анализа информации. Полноценная база знаний, помимо структурированной информации

о некоторой предметной области должна также содержать систему семантической (осмысленной) обработки информации и правила вывода, позволяющие делать автоматические умозаключения. Для таких задач подходит иерархический способ представления понятий и их отношений, являющейся одним из типов онтологии [1].

Основным источником информации для экспертной системы, основанной на базе знаний, являются данные, предоставляемые пользователем. Обычно наиболее удобной формой представления данных является текст на естественном языке. Таким способом можно предоставить широкий спектр информации в различных предметных областях. Однако для унифицированной работы с хранилищами данных с использованием компьютерных и программных систем необходимо использовать формализованные языки запросов. При работе с онтологией известным и хорошо зарекомендовавшим себя языком запросов является SPARQL [2]. Его использование рекомендовано W3C консорциумом, и он является одной из технологий semantic web [3]. Таким образом, актуальной задачей становится обработка текста на естественном языке для получения структурированных данных, удобных для построения запроса на языке SPARQL.

Кроме того, для эффективной работы с базой знаний требуется разработка определённой структуры онтологии, включающей некоторые ограничения на форму представления исходной информации о предметной области. Это позволяет унифицировать шаблоны SPARQL запросов для различных ситуаций, исходя из выделенных в исходном тексте типов высказываний и намерений. Способы построения онтологии предметной области является отдельной большой темой, тем не менее, некоторые аспекты структуры онтологии, принятые в разрабатываемой нами системе, также освещены в данной работе.

Практические аспекты применения рассматриваемой технологии – интеллектуальные чат-боты, экспертные системы, программы для интеллектуального анализа текста и автоматического предоставления выводов из него.

Следует отметить, что предлагаемая в данной работе система, предназначена для работы в первую очередь с грамматически и орфографически корректным текстом научно-технического стиля.

Анализ современных достижений в области обработки естественного языка при работе с онтологическими базами знаний

Различные подходы к построению запросов к данным на естественном языке разрабатываются уже на протяжении нескольких десятков лет, так как общение на таком языке является удобным для пользователей. Новые идеи применения естественно-языковых интерфейсов предлагаются постоянно. Далеко не все из них оказываются удачными и способны предложить хорошие решения. Однако любые новые исследования в данной области представляют ценность, так как они привносят новые идеи и позволяют лучше понять, что действительно работает, а что нет [4].

Важной проблемой экспертных систем с естественно-языковым интерфейсом является их зависимость от семантических грамматик, адаптированных к конкретной базе данных. При этом языки запросов на данный момент уже достаточно стандартизированы (например, SQL – для реляционной базы данных, SPARQL – для работы с онтологиями). Это позволяет отойти от привязки к внутреннему устройству базы данных [4].

К построению естественно-языкового интерфейса предлагался ряд различных подходов. Например, в работе [5] предлагается сосредоточиться на ограниченном наборе семантически трактуемых запросов с однозначным отображением отношений, атрибутов и значений, и использовать статистический семантический анализ. Авторы работ [6, 7] предложили модель, в которой, основываясь на естественно-языковом тексте, программа вводит описание проблемы и формирует код запроса. В работе [8] предложен метод обобщения предложений применительно к деревьям синтаксического разбора. Авторы работы [9] предложили систему преобразования, включающую три основных компонента: 1 – преобразование запроса на естественном языке в дерево запросов; 2 – интерактивная проверка преобразования (путем обращения к пользователю); 3 – SQL запрос.

В последние годы появляются методы, основанные на машинном обучении и нейронных сетях. Например, методика, применяющая обучение с подкреплением [10]. Для перевода вопросов на естественном языке в соответствующие запросы на SQL использовалась нейронная сеть. Однако, несмотря на огромный набор данных в обучающей выборке, точность модели оказалась не слишком высокой: точность выполнения – 59,4 %, точность логической формы – 48,3 %. Методика обучения также сильно зависит от предметной области. Для внедрения подобной системы потребуется обширная коллекция обучающих наборов, что зачастую обеспечить достаточно сложно.

Другой возможной сложностью при построении формальных запросов на основе естественного языка являются случаи омонимии и неоднозначности фраз пользователя. Кроме того, пользователь может использовать сленг и специфические термины предметной области, слова и выражения, имеющие в данном контексте специфическое значение, отличающееся от общеупотребительного. Подход к решению данной проблемы был предложен ещё в работах [11, 12]. Он заключается в уточняющих вопросах, адресуемых программой пользователю, с предложением пояснить, какой из вариантов значений он имел в виду.

Работы по созданию методов преобразования запросов на естественном языке в формальные запросы на SPARQL также проводились. Например, система LODQA (Linked Open Data Question Answering) [13] – программа, которая принимает запрос на естественном языке в качестве входных данных и возвращает SPARQL запросы и результат их выполнения. Система состоит из нескольких модулей. Первый модуль

отвечает за синтаксический анализ и создание графического представления запроса, который называется псевдографическим шаблоном. Как правило, узлы шаблона соответствуют основным именным группам, а связи – зависимостям между ними. Кроме того, псевдографический шаблон указывает, какой узел является фокусом запроса, то есть тем, что пользователь хочет получить в качестве ответа на запрос. Как только первый модуль сгенерировал псевдографический шаблон из заданного естественно-языкового запроса, включается модуль, отвечающий за поиск URI и значений узлов псевдографического шаблона. Из одного псевдографического шаблона может быть получено более одного привязанного шаблона путем нормализации. Третий модуль выполняет поиск в целевом наборе данных для соответствующих частей с учетом возможных изменений, которые могут возникнуть в наборе данных. Этот модуль пытается сгенерировать SPARQL запросы для всех возможных структурных вариаций. Затем SPARQL запросы отправляются на целевую конечную точку, где на них получаются ответы, которые направляются пользователю. Эти аргументы запроса могут быть примитивным типом. Для облегчения идентификации RDF-триплетов, слова в предложении лемматизируются и им присваиваются соответствующие грамматические характеристики. Рассмотренная система LODQA ориентирована на работу только с английским языком. Детальные особенности её функционирования в работе [13] не приводятся, ограничиваясь лишь общим описанием и анализом примеров работы.

Таким образом, мы видим, что, несмотря на достаточно давнее существование проблемы преобразования фраз на естественном языке в формальные запросы к базе данных, она не теряет свою актуальность, а новые подходы к её решению продолжают разрабатываться. Большинство работ посвящено построению на базе естественного языка SQL запросов (NL2SQL). Работы по исследованию преобразования естественного языка в запросы к онтологии относительно малочисленны и не раскрывают многих технических подробностей реализации подобной системы. Большинство программ для преобразования естественного языка в формальные запросы ориентировано на английский язык. Флективные языки, такие как украинский и русский имеют заметно отличающуюся структуру и требуют специфических методов анализа.

Предлагаемая модель анализа естественно-языкового текста для подготовки данных для запросов к онтологии

Онтология является формализацией некоторой области знаний при помощи концептуальной схемы, то есть связанной структуры данных. При этом онтологии хранятся в виде, удобном для компьютерной обработки [14]. Стандартным форматом хранения онтологии является OWL. Для построения формальных запросов к онтологии наибольшее распространение получил язык SPARQL. Более понятным и естественным для человека способом является представление информации на естественном языке. Поэтому актуальной становится задача перехода от фразы или набора фраз на естественном языке к пакету SPARQL запросов. При этом, информацию, полученную из онтологии, также имеет смысл во многих случаях оформлять в естественно-языковые фразы.

Онтологии, как правило, строятся так, чтобы соответствовать некоторой теме или предметной области. В системе может быть несколько онтологий. Для выбора наиболее подходящей онтологии может служить, например, отдельный файл, содержащий ключевые понятия каждой из имеющихся онтологий. Эти списки ключевых слов сопоставляются со словами из текстов. Выбирается для работы та онтология, для списка ключевых слов которой буден найден наибольший процент совпадений в анализируемом тексте.

Опишем последовательность анализа фразы на естественном языке флективного типа, к которым относятся такие языки как украинский и русский.

Первым этапом является графемный анализ – разбиение текстовой строки на отдельные предложения, на части сложных предложений и отдельные слова. Для этого в нашей системе используются инструменты библиотеки NLTK [15]. Слова представляются в виде программных объектов, способных также хранить характеристики слова, место в предложении и связи с другими словами. Использование такой структуры для представления слов упрощает реализацию этапов морфологического и синтаксического анализа.

Вторым этапом является выделение в предложениях маркерных слов для определения типа предложения. В украинском и русском языках главным критерием определении типа предложения (утвердительное или вопросительное) в письменном документе является наличие вопросительного знака в конце предложения. Определение более узкого подтипа высказывания идёт по наличию маркерных слов. Принятая классификация типов высказывания основана на работе [1]. В нашей модели, на данный момент, выделяются следующие простые подтипы: «нейтральное повествование» (маркерные слова отсутствуют), «время», «место», «причина», «объект», «способ», «направление», «потребность». Маркерных слов (или словосочетаний) может быть несколько. В таком случае определяется составной подтип, например, «время, место», «причина, время».

На следующем этапе анализа в предложениях выделяются именные группы. Именная группа – это набор связанных существительных и прилагательных, в совокупности описывающих некоторую сущность [16]. Главным словом именной группы является существительное. Все внешние связи, идущие к именной группе, рассматриваются как идущие к главному слову. Существительное, входящее в именную группу, считается связанным с главным словом именной группы в том случае, если оно стоит в родительном падеже и расположено, либо рядом с ним, либо разделено согласованными с главным или данным словом прилагательными или связанными с этими прилагательными наречиями (при наличии таковых).

Далее производится построение синтаксического графа для каждого из предложений. Именные группы выступают элементами этого графа наряду с отдельными словами. Сущность синтаксического графа –

представление связей между словами. Для флективных языков синтаксический граф можно построить исходя из характерных сочетаний частей речи в соответствующих словоформах. В нашей модели были приняты типы связей между словами, основанные на частях предложения. Наименование типа связи соответствует зависимому члену предложения, к которому строится данная связь. Прослеживается определённая связь этих типов с синтаксическими отношениями в словосочетании согласно [17]: атрибутивные, объектные, обстоятельственные, аппозитивные, комплетивные.

Под семантическим деревом высказывания будем понимать некоторый элементарный акт, совершаемый объектом или над объектом, или изменение какого-либо свойства объекта, которое происходит при определённых, указанных обстоятельствах. Структурно оно является подграфом синтаксического графа предложения, которому приписано соответствие одному из типов высказывания, выражаемых этим предложением [18]. При наличии маркерного слова оно становится корнем такого дерева. Главное понятие семантического дерева высказывания находится в непосредственной связи с маркерным словом. Связанные с главным понятием слова и слова, связанные с ними в соответствии с синтаксическим графом, образуют так называемые дополнительные обстоятельства семантического дерева высказывания. Например, рассмотрим предложение: «Где можно купить книги по программированию на Python?». Маркерным словосочетанием здесь будет «где можно», следовательно – это вопрос места, с дополнительным предикатом со значением соответствия (варианты со значением несоответствия были бы, например, «где нельзя», «где невозможно», «где запрещено»). Главным понятием будет глагол «купить». Дополнительными обстоятельствами будут слова «книги», «программирование» и «Python», связанные цепочкой связей, согласно синтаксическому графу.

При отсутствии маркерных слов, корнем семантического дерева высказывания будет абстрактное нулевое понятие, маркирующее нейтральность высказывания для данного дерева и не имеющего вербального представления в предложении. Оно служит лишь для унификации деревьев высказывания в данном программном представлении. Основным словом семантического дерева высказывания в этом случае может стать сказуемое или подлежащие. Тип высказывания при этом будет классифицирован как нейтральное высказывание (для повествовательного предложения) или общий вопрос (для вопросительного предложения). Связанные с ним слова и именные группы будут выступать как дополнительные обстоятельства.

Семантические деревья высказывания используются для построения запросов к онтологии на языке SPARQL. При этом используются заранее известные шаблоны запросов, вид которых зависит от типа высказывания. Для создания шаблонов запросов важную роль играет принятая в данной системе структура онтологии. Работа программы, реализующая вышеописанную модель разбора может быть представлена в виде UML-диаграммы деятельности, показана на рисунке.

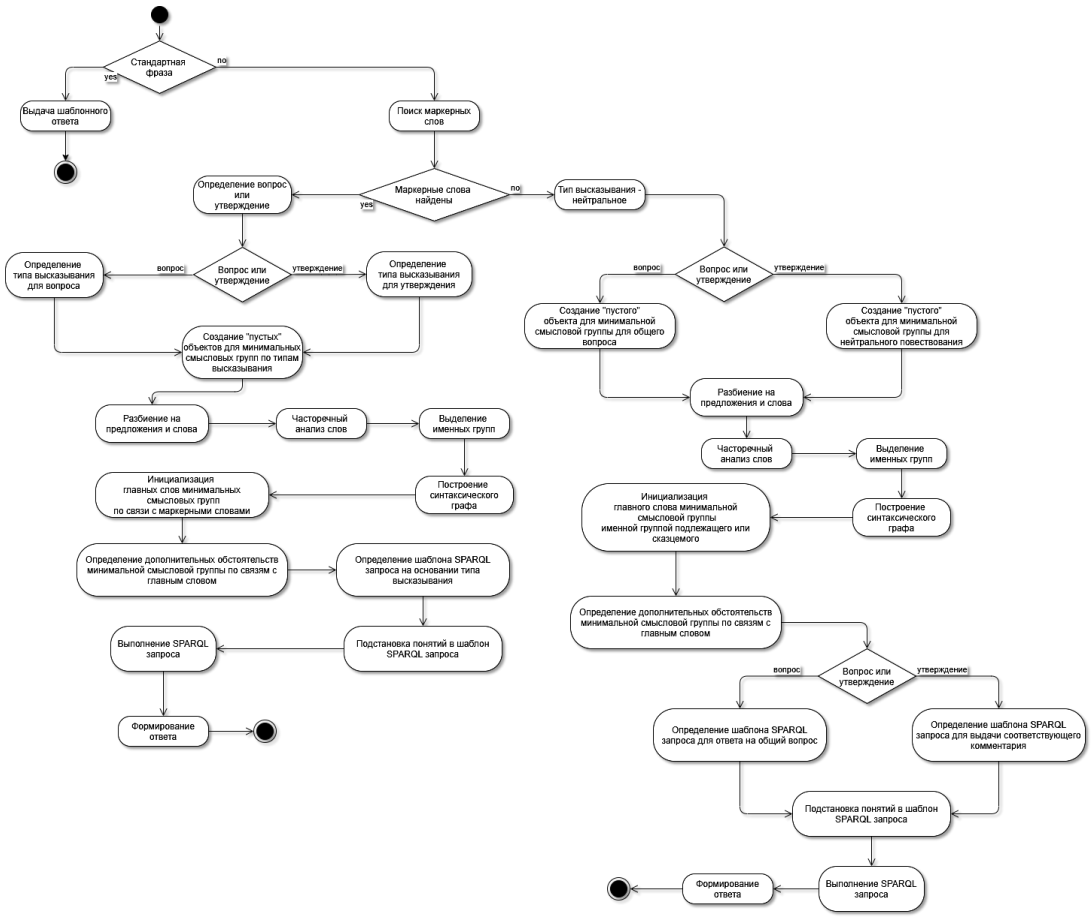

Рисунок. UML-диаграмма деятельности, иллюстрирующая общую схему работы системы

Краткая характеристика структуры онтологии, используемой в разрабатываемой системе

Концепция строения онтологии, принятая на данный момент в нашей системе следующая. Заданы классы верхнего уровня иерархии, классифицирующие понятия, входящие в онтологию: действие (action), причина (cause), метод (method), предмет (object), место (place), время (time). Возможны и другие варианты, в соответствии с выделяемыми типами высказываний. Эти классы делятся на более узкие подклассы, но всё же достаточно абстрактные и объемные. Например, класс действие (action) подразделяется на активное действие и пассивное действие. Активное действие обозначает именно деятельность, непосредственное выполнение некоторых функций («организовать», «разрабатывать», «начинать», «проводить»). Пассивное действие – это действия, выражаемые глаголами, но не подразумевающие совершения действия, а скорее характеризующие объекты и их состояние («состоять», «существовать», «заключаться»). Идущие ниже по иерархии понятия могут быть представлены сколь угодно глубоко вложенными классами.

Используемая в нашей системе структура онтологии не предполагает наличия экземпляров объектов. Ограничение в онтологии только классами для обозначения понятий унифицирует SPARQL запросы. В онтологии также определяются свойства. Они задают неиерархические связи между понятиями. Согласно стандарту OWL, свойства имеют Domain и Range. В предлагаемой структуре онтологии раздел Domain состоит из пересечения набора независимых понятий, раздел Range представляет набор понятий, являющихся следствием комбинации понятий, представленных в Domain. Таким образом свойства являются своеобразными функциями, где независимые переменные (факторы) представлены в Domain, а зависимые (отклики) – в Range. Например, в Domain записано: «начинать», «разрабатывать», «концепция информатизации», «Украина», в Range записано: «шестидесятые годы». При этом понятие «шестидесятые годы» является наследником класса «время (time)». Указанное свойство выражает то, что концепцию информатизации в Украине начали разрабатывать в шестидесятые годы. Принадлежность понятий в Domain имеет значение для верной интерпретации сути свойства. В данном случае понятия «начинать», «разрабатывать» – это действия, а «Украина» – место.

Принцип построения и примеры SPARQL запросов

Как уже отмечалось выше, на основе семантических деревьев высказываний, выделенных при анализе исходной фразы, формируются SPARQL запросы. Структура запроса зависит от типа высказывания, лежащего в основе данного семантического дерева высказывания. Также возможна более тонкая настройка вариантов SPARQL запросов в пределах каждого типа. Рассмотрим некоторые примеры шаблонов SPARQL запросов в зависимости от типа высказывания в семантическом дереве высказывания.

Нейтральное высказывание не предполагает конкретного ответа, но его можно прокомментировать, найдя связанные понятия. В этом случае, если в семантическом дерева высказывания нет дополнительных обстоятельств, шаблон SPARQL запроса может выглядеть следующим образом:

```
SELECT DISTINCT ?res WHERE {
?y rdfs:domain :inst_name.
?y rdfs:range ?z.
?z rdfs:label ?res.}
```

В шаблоне *inst_name* – переменная, сформированная из главного понятия семантического дерева высказывания, таким образом, чтобы соответствовать именованию классов, принятому в онтологии (составляются из набора основ слов, входящих в главное понятие). Запрос также предлагает вывести label – поле в котором понятия записаны на более понятном человеку языке (названия классов предполагают слитное написание комбинации основ).

Если семантическое дерево высказывания имеет также дополнительные обстоятельства, шаблон запроса усложняется:

```
SELECT DISTINCT ?res WHERE {
?y rdfs:domain :inst_name.
?y rdfs:domain :cur_sup_name_1.
…
?y rdfs:domain :cur_sup_name_n.
?y rdfs:range ?z.
?z rdfs:label ?res.}
```

В шаблоне *inst_name*, как и в предыдущем случае, – переменная сформированная из главного понятия семантического дерева высказывания; *cur_sup_name_1* … *cur_sup_name_n* – переменные, сформированные из дополнительных обстоятельств.

Следующий пример иллюстрирует SPARQL запрос для случая семантического дерева высказывания причины:

```
SELECT DISTINCT ?res WHERE {
?x0 rdfs:subClassOf :cause.
?x_n rdfs:subClassOf ?x0.
…
?last_x rdfs:subClassOf ?x_n.
:inst_name rdfs:subClassOf last_x.
?y rdfs:domain :inst_name.
?y rdfs:domain :sup_1.
…
?y rdfs:domain :sup_i.
?y rdfs:range ?z.
?z rdfs:label ?res.}
```

В шаблоне имеются следующие внешние параметры: *inst_name* – переменная, сформированная из главного понятия семантического дерева высказывания; *sup_1* … *sup_n* – переменные, сформированные из дополнительных обстоятельств. Такая структура шаблона запроса гарантирует, что основное понятие семантического дерева высказывания будет рассмотрено в контексте причины вне зависимости от глубины иерархии конкретного понятия.

Следующий шаблон SPARQL запроса подойдёт для семантического дерева высказывания времени:

```
SELECT DISTINCT ?res WHERE {
?y rdfs:domain :inst_name.
?y rdfs:domain :sup_1.
…
?y rdfs:domain :sup_i.
?x rdfs:subClassOf :time.
?y rdfs:range ?z.
?z rdfs:subClassOf ?x.
?z rdfs:label ?res. }
```

В шаблоне имеются следующие внешние параметры: *inst_name* – переменная, сформированная из главного понятия семантического дерева высказывания; *sup_1* … *sup_n* – переменные, сформированные из дополнительных обстоятельств. Здесь упрощённо полагается, что все классы, обозначающие время, наследуются от класса `time` непосредственно.

Шаблоны SPARQL запросов для семантических деревьев высказываний других типов в целом близки по структуре к вышеуказанным и здесь не приводятся. С другой стороны, это показывает достаточную универсальность предложенной модели для работы с высказываниями различных типов.

Результатом SPARQL запроса является набор понятий, отвечающий его условиям отбора. Если подходящих объектов в онтологии нет, то возвращается пустой список. В случае отсутствия данных, программа выполняет новые SPARQL запросы с использованием редуцированных именных групп. Ответ в таком случае может оказаться менее релевантным, но всё же, некоторый ответ на запрос будет получен всегда.

Полученный в результате запроса набор понятий, безусловно, существенен, однако для дальнейшей машинной обработки и составления естественно-языковой фразы этой информации недостаточно. Поэтому производится уточнение, наследником какого именно класса более высокой иерархии является каждое из найденных понятий. С этой целью для каждого из полученных понятий выполняется серия SPARQL запросов, направленных на поиск класса, являющегося родителем класса, используемого в предыдущей итерации. Итерации завершаются на этапе, когда родителем класса на данной итерации станет класс Thing, являющийся корневым для всех классов. Шаблон такого запроса приведен ниже:

```
SELECT DISTINCT ?res WHERE {
?x rdfs:label current_class.
?x rdfs:subClassOf ?y.
?y rdfs:label ?res. }
```

В шаблоне *current class* – подставляемое значение параметра label для класса из предыдущей итерации, на первой итерации – это понятие из выданных основным запросом.

Таким образом, мы получаем цепочку иерархии понятий более высокого уровня, к которым относится отобранное основным запросом понятие. Эта информация полезна для более информативного и корректного построения ответа на естественном языке на основе найденных понятий, удовлетворяющих основному запросу. Так зная, является ли данное понятие предметом, действием, временем, местом и т. п., нам будет легче расставить слова в надлежащем порядке и, при необходимости, связать их соответствующими предлогами.

Приведём пример анализа фразы, составления запроса и получения ответа из онтологии. В примере использована онтология, составленная по докладу академика В.С. Михалевича о концепции информатизации общества. Вопрос пользователя: «В чём состоит значение информатизации для человеческого общества?». Из этой фразы система выделяет одно семантическое дерево высказывания: тип высказывания – «вопрос – запрос перечня объектов»; маркерное словосочетание – «в чём состоит»; главное понятие – именная группа «значение информатизации»; дополнительные обстоятельства – именная группа «человеческое общество». Для данного случая предусмотрен SPARQL шаблон, в который подставляются лемматизированные понятия (выделены курсивом и подчёркиванием):

```
SELECT DISTINCT ?res WHERE {
?y rdfs:domain :ЗначенИнформатизац.
?y rdfs:domain :ЧеловеческОбществ.
?y rdfs:range ?z.
?z rdfs:label ?res. }
```

Это достаточно простой случай, при котором есть только одно дополнительное обстоятельство, а понятия специфически не маркированы, чтобы быть наследниками определённых классов. Результат, возвращаемый запросом – набор понятий (именных групп): «решение научно-технических проблем», «внесение значительного экономического вклада», «повышение производительности труда».

Полученные наборы понятий могут быть использованы в дальнейшем. Например, если приложение является чат-ботом, то наборы понятий наряду с семантическими данными и историей диалога могут быть использованы при формировании ответа. Однако подробное описание дальнейшей обработки результатов SPARQL запроса выходит за рамки данной статьи.

Выводы и перспективы дальнейших исследований

Разработан метод анализа естественно-языкового текста на украинском и русском языках, позволяющий строить семантические деревья высказываний, удобные для формирования запросов к онтологии на языке SPARQL. Семантические деревья высказываний характеризуются маркерными словами и типом высказывания. Из исходного предложения может быть выделено несколько семантических деревьев высказываний, для каждого из которых будет построен свой SPARQL запрос.

Представлены шаблоны для построения SPARQL запросов различных типов к онтологии естественно-языкового документа. Вид шаблона определяется типом высказывания в соответствии с семантическим деревом высказывания. Шаблон содержит подставляемые параметры, которые получаются из соответствующих вершин семантического дерева высказывания.

В модели использован подход редукции именных групп при неудовлетворительном результате SPARQL запроса (отсутствие данных на выходе). Это позволяет запросить информацию о более общих понятиях, и получить некоторый ответ, хотя, возможно, и менее релевантный.

Предложенная система может случить элементом дружественного естественно-языкового интерфейса при работе с базой знаний, представленной в виде онтологии или набора онтологий, также она может использоваться при создании интеллектуальных чат-ботов.

Дальнейшими перспективами развития данной программной системы является расширение и более тонкая классификация высказываний пользователя, моделирование построения уточняющих вопросов при возникновении неразрешимых неоднозначностей, генерация грамматически верных фраз в качестве ответа, вид которых зависит от семантики вопроса.

Литература

1. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер. 2000. 384 с.
2. Антониоу Г. Семантический веб. М.:: ДМК-Пресс. 2016.240 с.
3. SPARQL 1.1 Query Language [Электронный ресурс]. W3C. 2013. Режим доступу до ресурсу: https://www.w3.org/TR/sparql11-query/.
4. Galitsky B. Developing Enterprise Chatbots. Learning Linguistic Structures. San Jose: Springer. 2019. 559 p.

5. Popescu A.M., Etzioni O., Kautz H.A. Towards a theory of natural language interfaces to databases. *IUI*. 2003. С. 149–157.
6. Galitsky B., Usikov D. Programming Spatial Algorithms in Natural Language. *AAAI Workshop Technical Report WS-08-11*. 2008. P. 16–24.
7. Quirk C., Mooney R., Galley M. Language to code: learning semantic parsers for if-this-then-that recipes. *ACL*. 2015. P. 878–888.
8. Galitsky B., De La Rosa J.L., Dobrocsi G. Mapping syntactic to semantic generalizations of linguistic parse trees. Proceedings of the twenty-fourth international Florida artificial intelligence research society conference. 2011. P. 168–173.
9. Li F., Jagadish H.V. Understanding natural language queries over relational databases. *SIGMOD Record*. 2016. N 45. P. 6–13.
10. Zhong V., Xiong G., Socher R. Seq2SQL: generating structured queries from natural language using reinforcement learning. [Электронный ресурс]. 2017. Режим доступу до ресурсу: https://arxiv.org/pdf/1709.00103.pdf.
11. Kupper D., Strobel M., Rosner D. Nauda – a cooperative, natural language interface to relational databases. *SIGMOD conference*. 1993. P. 529–533.
12. Li Y., Yang H., Jagadish H.V. Nalix: an interactive natural language interface for querying xml. *SIGMOD conference*. 2005. P. 900–902.
13. Shaik S., Kanakam P., Hussain S.M., Suryanarayana D. Transforming Natural Language Query to SPARQL for Semantic Information Retrieval. *International Journal of Engineering Trends and Technology*. 2016. N 7. P. 347–350.
14. Лапшин В. А. Онтологии в компьютерных системах. Москва: Научный мир. 2010. 224 с.
15. Natural Language Toolkit. NLTK 3.4.5 documentation. [Электронный ресурс]. NLTK Project. 2019. Режим доступу до ресурсу: https://www.nltk.org.
16. Crystal D. A Dictionary of Linguistics and Phonetics. 2008. 560 p.
17. Курышева М.В. Русский язык: синтаксический анализ словосочетания и простого предложения. Томск: Томский государственный педагогический университет, 2014.
18. Шелманов А. О. Исследование методов автоматического анализа текстов и разработка интегрированной системы семантико-синтаксического анализа: дис. канд. техн. наук: 05.13.17. Москва. 2015. 210 с.

References

1. Gavrilova T.A., V.F. Khoroshevsky (2000) *Knowledge Base of Intelligent Systems*. St. Petersburg: Peter.
2. Antoniou G. (2016) *Semantic Web*. Moscow: DMK-Press.
3. W3C (2013) *SPARQL 1.1 Query Language* [Online] Available from: https://www.w3.org/TR/sparql11-query/ [Accessed: 11 February 2020].
4. Galitsky B. (2019) *Developing Enterprise Chatbots. Learning Linguistic Structures*. San Jose: Springer.
5. Popescu A. M., Etzioni O., Kautz H. A. (2003) Towards a theory of natural language interfaces to databases. *IUI*. p. 149–157.
6. Galitsky B., Usikov D. (2015) Programming Spatial Algorithms in Natural Language. *AAAI Workshop Technical Report WS-08-11*. P. 16–24.
7. Quirk C., Mooney R., Galley M. (2015) Language to code: learning semantic parsers for if-this-then-that recipes. *ACL*. P. 878–888.
8. Galitsky B., De La Rosa J.L., Dobrocsi G. (2011) Mapping syntactic to semantic generalizations of linguistic parse trees. *Proceedings of the twenty-fourth international Florida artificial intelligence research society conference*. P. 168–173.
9. Li F., Jagadish H. V. (2016) Understanding natural language queries over relational databases. *SIGMOD Record*. 45. P. 6–13.
10. Zhong V., Xiong G., Socher R. (2017) *Seq2SQL: generating structured queries from natural language using reinforcement learning.* [Online] Available from: https://arxiv.org/pdf/1709.00103.pdf [Accessed: 11 February 2020].
11. Kupper D., Strobel M., Rosner D. (1993) Nauda – a cooperative, natural language interface to relational databases. *SIGMOD conference*. P. 529–533.
12. Li Y., Yang H., Jagadish H. V. (2005) Nalix: an interactive natural language interface for querying xml. *SIGMOD conference*. P. 900–902.
13. Shaik S., Kanakam P., Hussain S.M., Suryanarayana D. (2016) Transforming Natural Language Query to SPARQL for Semantic Information Retrieval. *International Journal of Engineering Trends and Technology*. 7. P. 347–350.
14. Lapshin V.A. (2010) *Ontologies in computer systems*. Moscow: Scientific World.
15. NLTK Project (2019) *Natural Language Toolkit. NLTK 3.4.5 documentation.* [Online] Available from: https://www.nltk.org [Accessed: 11 February 2020].
16. Crystal D.A (2008) *Dictionary of Linguistics and Phonetics* Wiley-Blackwell.
17. Kurysheva M.V. (2014) *Russian language: syntactic analysis of phrases and simple sentences*. Tomsk: Tomsk State Pedagogical University.
18. Shelmanov A.O. (2015) Ph.D. Tresses: Study of methods for automatic text analysis and development of an integrated system of semantic-syntactic analysis. Moscow.

Получено 02.03.2020

Об авторах:

[1]*Литвин Анна Андреевна*,
аспирантка Института кибернетики имени В.М. Глушкова НАН Украины.
Количество научных публикаций в украинских изданиях – 2.
Количество научных публикаций в зарубежных изданиях – 1.
http://orcid.org/0000-0002-5648-9074,

[1]*Величко Виталий Юрьевич*,
кандидат технических наук, доцент,
старший научный сотрудник отдела микропроцессорной техники
Института кибернетики имени В. М. Глушкова НАН Украины.

Количество научных публикаций в украинских изданиях – 75.
Количество научных публикаций в зарубежных изданиях – 27.
H-index: Google Scholar – 11
Scopus – 1,
http://orcid.org/0000-0002-7155-9202,

[2]*Каверинский Владислав Владимирович*,
кандидат технических наук,
старший научный сотрудник отдела износостойких
и коррозионностойких порошковых конструкционных материалов
Института проблем материаловедения им. И.Н. Францевича НАН Украины.
Количество научных публикаций в украинских изданиях – 81.
Количество научных публикаций в зарубежных изданиях – 18.
H-index: Google Scholar – 4.
Scopus – 2.
http://orcid.org/0000-0002-6940-579X.

Место работы авторов:

[1]Институт кибернетики имени В.М. Глушкова НАН Украины.
03187, Киев-187, проспект Академика Глушкова, 40.
E-mail: litvin_any@ukr.net,
aduisukr@gmail.com

[2]Институт проблем материаловедения им. И.Н. Францевича НАН Украины.
03142, Киев, ул. Кржижановского, 3.
E-mail: insamhlaithe@gmail.com