Н. А. Леонов, И. Г. Захарова
Тюменский государственный университет, г. Тюмень
УДК 004.82

МЕТОДЫ И АЛГОРИТМЫ ДЛЯ ПОСТРОЕНИЯ ОНТОЛОГИИ ДЛЯ QA-СИСТЕМЫ

Аннотация. В статье рассмотрена проблема автоматического построения онтологии предметной области и ее обновления при добавлении новых знаний для вопросно-ответной системы. Произведен анализ существующих подходов и реализован метод построения онтологии из неструктурированного текста электронной книги.

Ключевые слова: онтология, хранение знаний, вопросно-ответные системы, объединение онтологий, извлечение концептов из текста.

Введение. Зачастую для оперативной валидной обратной связи в таких сферах как образование, телемедицина, банковская сфера и др. принято использовать вопросно-ответные системы (QA-системы). Однако традиционная QA-система, как правило, характеризуется отсутствием следующих важных свойств:

- Мониторинг данных о ситуации (у кого, где, когда возник вопрос).
- Адаптивность ответа.
- Дообучение (исключением являются модели типа GPT, но в них нет валидации ответов).

В рамках решения задачи построения вопросно-ответной системы первичным этапом является построение базы знаний. Знания в вопросно-ответной системе могут храниться в различных форматах:

- база данных;
- текстовые файлы;
- онтологии.

Именно от того, каким образом знания будут храниться в вопросно-ответной системе, и будет напрямую зависеть качество и полнота ответов. Онтологии используются в вопросно-ответных системах для определения семантической близости между запросом пользователя и базой знаний. Онтология позволяет системе понимать смысл запроса пользователя и находить наиболее релевантные ответы в базе знаний.

Вопросам автоматического построения онтологии посвящено достаточное число исследований. В работе [1] предложен подход к построению онтологии предметной области на основе алгоритмов машинного обучения и подходов обработки естественного языка. Данный подход позволяет более детально и объемно строить модель представляемых знаний, что в свою очередь позволяет проводить более качественный семантический анализ.

В статье [2] показан пример разработки автоматически генерируемой онтологии компьютерных наук (CSO), которая включает в себя около 26 тыс. тем и 226 тыс. семантических связей. Данная онтология была создана путем применения алгоритма Klink-2 к очень большому набору данных из 16 миллионов научных статей. Особенностью данной онтологии является то, что ее можно обновлять автоматически, запустив Klink-2 для последних корпусов публикаций.

Процесс извлечения объектов из текстовых документов для пополнения онтологии может быть разделен на два этапа.

1. Извлечение цепочек символов — идентификация их как объектов онтологии, с последующей классификацией для отнесения цепочки к той или иной известной семантической категории.

2. Отображение полученных объектов на известные словари и базы знаний для определения их концепта. Объекты, которые не были отнесены ни к одному из имеющихся концептов, могут стать кандидатами на новые концепты [3].

Первый этап извлечения объектов можно выполнить несколькими способами, к основным из которых относят: статистические методы извлечения именованных сущностей [4], методы, основанные на грамматиках [5], а также методы, основанные на извлечении часто встречающихся цепочек токенов (к группе таких методов можно отнести алгоритмы типа GSP [6], PrefixSpan [7] или SPAM [8]).

Для второго этапа работы алгоритма полезно применять словари синонимов, алгоритмы разрешения кореферентности и алгоритмы поиска с опечатками в словарях. Например, в качестве такого словаря можно использовать Wikipedia [9].

Основная группа отношений для онтологии — отношения типа «is-a», то есть отношения, задающие иерархию концептов. Популяр-

ным методом извлечения отношений подобного рода является метод, основанный на лексико-синтаксических шаблонах [10]. Также для решения данной задачи можно использовать методы синтаксического анализа и, получая дерево синтаксического разбора, извлекать из него необходимые отношения.

Процесс извлечения других семантических отношений не отличается от процесса извлечения иерархических отношений.

Для пополнения базы знаний потребуется разработать алгоритм объединения нескольких онтологий. Для этого существует несколько методов, включая:

1. Автоматическое объединение. Эти алгоритмы могут использовать различные методы оптимизации, такие как эволюционные алгоритмы, генетические алгоритмы или методы математической оптимизации.

2. Использование онтологических соответствий. Этот метод включает в себя использование существующих соответствий между онтологиями для автоматического объединения. Например, если две онтологии имеют схожие концепты, свойства или отношения, то можно использовать соответствие между ними для автоматического объединения.

3. Использование онтологических выравниваний. Онтологические выравнивания позволяют определить соответствия между концептами, свойствами и отношениями из разных онтологий, что позволяет автоматически объединять их.

Исходя из проведенного анализа можно сделать вывод о том, что тема исследования актуальна, востребована и значима.

Основная цель работы — исследование, разработка и программная реализация технологии конструирования и обновления базы знаний предметной области в виде онтологии для QA-системы (на примере сопровождения онлайн курса). В рамках проекта были поставлены следующие задачи:

1. Исследовать существующие подходы построения онтологии предметной области на основе неструктурированного текста.

2. Выполнить сравнение различных подходов автоматического построения онтологии.

3. Сконструировать базу знаний предметной области на основе материалов электронного курса по изучению дисциплины “Современные системы управления базами данных”.

4. Исследовать подходы объединения нескольких онтологий для дополнения базы знаний предметной области новыми знаниями.

Задачи построения и объединения онтологий. Общая постановка задачи построения онтологии из текста может быть сформулирована следующим образом:

Дан набор текстовых документов $D = \{d_1, d_2, \ldots, d_n\}$ и множество категорий $C = \{c_1, c_2, \ldots, c_m\}$. Требуется построить онтологию $O = (C, R)$, где C — множество категорий, а R — множество отношений между категориями на основе анализа текстовых документов D.

Практически во всех сферах применения вопросно-ответных систем вопросы можно разделить на 3 категории: вопросы типа «что», вопросы типа «как» и вопросы типа «почему» (табл. 1). Предварительно для более релевантного поиска ответа на вопрос будет производиться классификация по его типу.

Таблица 1

Типы вопросов в различных сферах

Сфера/ Тип вопроса	*Что?*	*Как?*	*Почему?*
Банковская	Что такое инвестиционный счет?	Как перевести деньги в другой банк?	Почему сняли комиссию за перевод?
Юридическая	Какое наказание за езду без страховки?	Как составить заявление о краже?	Почему было применено наказание, а не штраф?
Медицинская	Что такое HGB в анализе крови?	Как лечить простуду?	Почему болит голова?
Техническая	Что такое класс в Java?	Как переопределить метод родительского класса?	Почему выдает ошибку NullPointerException?
Образовательная	Что такое первичный ключ?	Как сделать группировку в SQL?	Почему не выполняется запрос group by?

Для построения общей онтологии потребуется реализовать метод объединения нескольких баз знаний для разных типов вопросов (из учебника, из методических указаний, из технической документации и т. д.).

Общая постановка задачи объединения нескольких онтологий может быть сформулирована как:

• Пусть имеется набор из N исходных онтологий $O_1, O_2, \ldots, O_N$, где каждая онтология представляет собой набор концептов, свойств и отношений между ними.

• Требуется построить единую онтологию O, которая содержит все знания из исходных онтологий и не содержит дубликатов и противоречий.

Отображение онтологии из онтологии $O_1 = (S_1, C_1)$ в $O_2 = (S_2, C_2)$ считается морфизмом $f: S_1 \to S_2$ онтологических сигнатур таким образом, что $C_2 = f(C_1)$, т. е. все интерпретации, которые удовлетворяют аксиомам O_2, также удовлетворяют переведенным аксиомам O_1.

Формально задача объединения онтологий может быть поставлена как задача оптимизации, где целью является построение новой онтологии O, которая удовлетворяет всем ограничениям и максимизирует качество и полноту знаний. Для решения этой задачи могут использоваться различные методы оптимизации, такие как эволюционные алгоритмы, генетические алгоритмы или методы математической оптимизации.

Материалы и методы. Для предобработки текста и извлечения именованных сущностей использовались библиотеки Nltk и spacy. Для обучения была задействована модель ru_core_news_lg [11].

Для извлечения триплетов вида субъект-глагол-объект использовалась функция subject_verb_object_triples библиотеки textacy. Затем для удобного хранения полученных данных в объекте DataFrame и дальнейшей его обработки использовалась библиотека pandas.

Для визуального представления полученной онтологии и визуализации подграфов после выполнения фильтрации по нескольким концептам была использована библиотека pyplot.

Результаты. По результатам проведенного эксперимента по извлечению триплетов вида субъект-глагол-объект из текста книги “Освой самостоятельно SQL за 10 минут” автора Бена Форта была получена онтология, представленная на рис. 1.

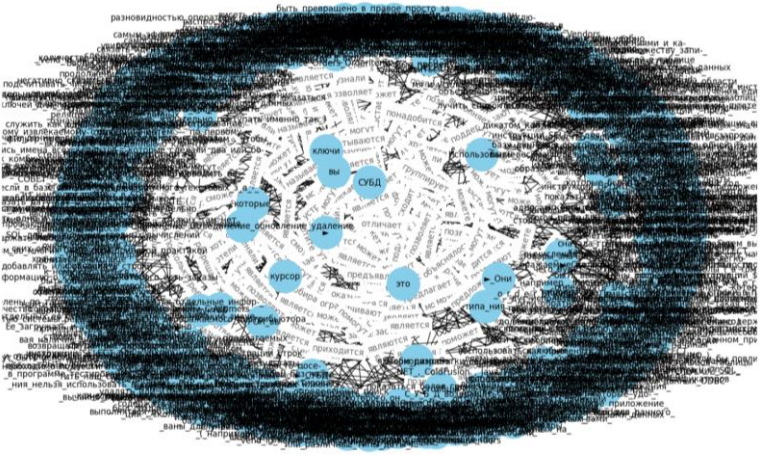

Рис. 1. Построенная онтология в общем виде

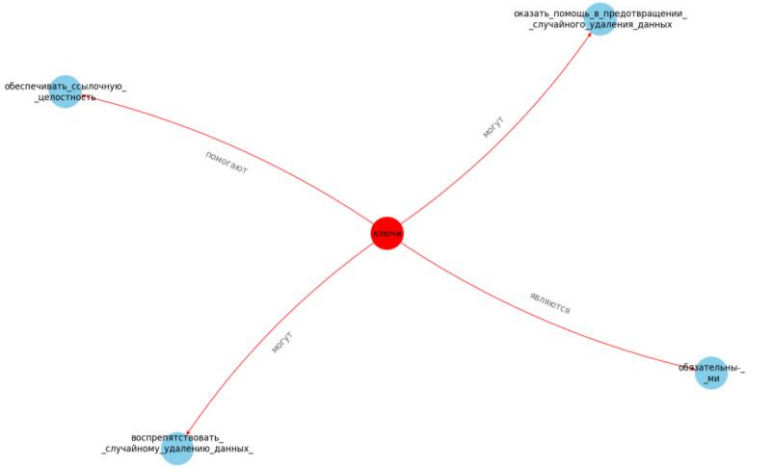

Рис. 2. Типы вопросов

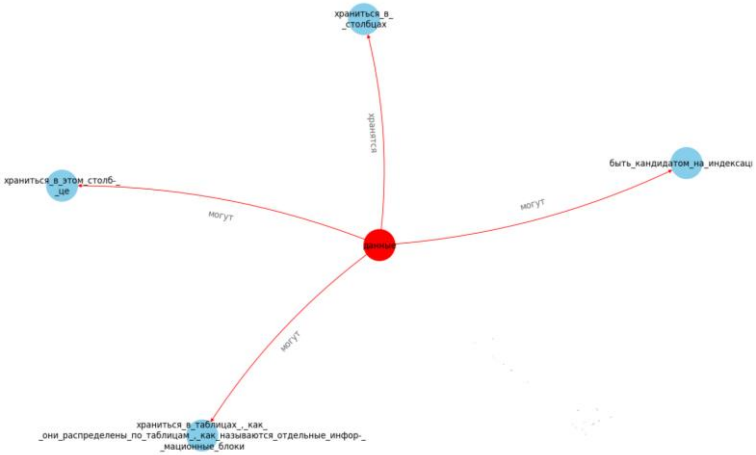

Рис. 3. Типы вопросов

Заключение. В целом, для автоматического построения онтологии предметной области существует множество подходов, анализ которых и программная реализация одного из них было представлены в данной работе.

Для более репрезентабельного представления была проведена фильтрация по концептам «ключи» и «данные», результаты которой представлены на рис. 2 и 3 соответственно.

Использование онтологии в вопросно-ответной системе позволяет учитывать контекст текущего диалога и выдавать пользователю наиболее релевантные ответы на его вопрос, а методы обновления онтологии позволяют поддерживать базу знаний в актуальном состоянии, за счет чего система может расширяться под любые потребности.

СПИСОК ЛИТЕРАТУРЫ

1. Konys A. Knowledge repository of ontology learning tools from text / A. Konys. — Direct text // Procedia Computer Science. — 2019. — V. 159. — Pp. 1614-1628.
2. The computer science ontology: a large-scale taxonomy of research areas / A. A. Salatino, T. Thanapalasingam, A. Mannocci [et al.]. — Direct text //

The Semantic Web–ISWC 2018: 17th International Semantic Web Conference. — Monterey : Springer International Publishing, 2018. — Pp. 187-205.
3. Платонов А. В. Методы автоматического построения онтологий / А. В. Платонов, Е. А. Полещук. — Текст : непосредственный // Программные продукты и системы. — 2016. — № 2 (114). — С. 47-52.
4. Nugumanova A. Applying the latent semantic analysis to the issue of automatic extraction of collocations from the domain texts / A. Nugumanova, I. Bessmertny. — Direct text // Knowledge Engineering and the Semantic Web: 4th International Conference, KESW 2013. — St. Petersburg : Springer Berlin Heidelberg, 2013. — Pp. 92-101.
5. Jurafsky D. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition / D. Jurafsky, J. H. Martin. — Second Edition. — Upper Saddle River : Prentice Hall, 2008. — 628 с. — Текст : непосредственный.
6. Aggarwal C. C. Frequent pattern mining algorithms: A survey / C. C. Aggarwal, M. A. Bhuiyan, M. A. Hasan. — Direct text // Springer International Publishing. — 2014. — Pp. 19-64.
7. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth / J. Han, J. Pei, B. Mortazavi-Asl [et al.]. — Direct text // Proceedings of the 17th international conference on data engineering. — IEEE, 2001. — Pp. 215-224.
8. Sequential pattern mining using a bitmap representation / J. Ayres, J. Flannick, J. Gehrke, T. Yiu. — Direct text // Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. — 2002. — Pp. 429-435.
9. Heist N. Entity extraction from Wikipedia list pages / N. Heist, H. Paulheim. — Direct text // The Semantic Web: 17th International Conference, ESWC 2020. — Heraklion : Springer International Publishing, 2020. — Pp. 327-342.
10. Biperpedia: an ontology for search applications / R. Gupta, A. Halevy, X. Wang [et al.]. — Direct text // Proceedings of the VLDB Endowment. — 2014. — № 7. — С. 505–516.
11. Russian spaCy Models Documentation. — Текст : электронный // spaCy · Industrial-strength Natural Language Processing in Python : [сайт]. — URL: https://spacy.io/models/ru#ru_core_news_lg (дата обращения: 16.05.2023).