Д. А. Евсеев, Генерация запросов для ответа на сложные вопросы на русском языке с использованием синтаксического парсера, *Искусственный интеллект и принятие решений*, 2021, выпуск 3, 57–65

DOI: 10.14357/20718594210305

Генерация запросов для ответа на сложные вопросы на русском языке с использованием синтаксического парсера

Д. А. Евсеев

Московский физико-технический институт, г. Москва, Россия

Аннотация. В работе описывается система, которая переводит вопрос на естественном языке в SPARQL-запрос. В состав вопросно-ответной системы входят: синтаксический парсер, который строит синтаксическое дерево предложения; компонент, определяющий шаблон SPARQL-запроса по синтаксическому дереву; модели, находящие сущности и отношения, которые должны быть вставлены в слоты шаблона SPARQL-запроса. Для извлечения сущностей и ранжирования возможных отношений используется BERT. Одна из особенностей обучения BERT для вопросно-ответной системы на русском языке состоит в малом количестве данных. В связи с этим, в работе исследуется обучение мультиязычного BERT, предобученного на датасете LC-QUAD2.0 задачам извлечения сущностей и ранжирования отношений на малом количестве русских данных из датасета RuBQ. Вопросно-ответная система показывает на датасете RuBQ более высокую точность ответов на вопросы, чем предыдущие подходы.

Ключевые слова: вопросно-ответная система, база знаний, генерация запросов, мультиязычный BERT.

DOI 10.14357/20718594210305

Введение

Вопросно-ответная система получает вопрос на естественном языке, обрабатывает его и генерирует ответ. Вопросно-ответные системы активно используются в виртуальных ассистентах (Алиса, Amazon Alexa, и т. д.). Вопросно-ответные системы могут использовать в качестве источников набор текстовых документов [1-3] или базы знаний [4,5]. Преимущество баз знаний состоит в структурированном представлении фактов по сравнению с текстом, более удобном для ответа на вопросы, ответ на которые получается путем объединения нескольких фактов. Кроме того, представление фактов в виде графа знаний, где узлы – сущности, а ребра – отношения между сущностями, не зависит от языка (к какому-то определенному языку относятся только названия сущностей и отношений). Базу знаний можно также представить как набор триплетов (subject_entity, relation, object_entity), например, Россия, столица, Москва. Для ответа на вопрос по базе знаний (KBQA) нужно определить вершины, которые соответствуют сущностям из вопроса, и путь в графе от этих сущностей до сущности-ответа. Чтобы ответить на вопрос, может требоваться извлечение одного триплета из базы знаний (простой вопрос) или нескольких триплетов, а также операции объединения, пересечения множеств сущностей (сложный вопрос).

SPARQL-запрос включает в себя сущности из базы знаний и команды для извлечения из нее искомой сущности. Один из подходов к задаче ответа на вопросы по базе знаний — перевод вопроса на естественном языке в

✉ *Евсеев Дмитрий Андреевич. E-mail: dmitrij.euseew@yandex.ru*

SPARQL-запрос, выполнение которого дает ответ. В данной статье описывается подход для построения шаблона SPARQL-запроса с использованием синтаксического парсера.

KBQA-система, описываемая в данной статье, состоит из следующих компонентов: построение шаблона SPARQL-запроса, извлечение сущностей из вопроса, связывание сущностей с базой знаний, ранжирование отношений. В настоящее время во многих задачах обработки естественного языка (в том числе, и в задаче извлечения сущностей) наиболее высокие метрики дает подход, основанный на предобучении языковой модели на большом корпусе, а затем донастройка весов модели на конкретной задаче [6]. Модели на основе BERT показвают SOTA-результаты на датасетах CoNLL-2003 [6] и OntoNotes [7].

Датасет RuBQ [8] содержит вопросы на русском языке и соответствующие SPARQL-запросы. В RuBQ 300 вопросов для валидации моделей и 1200 вопросов для тестирования. Одна из целей данной работы – исследовать, какое качество ответа на вопросы будет у модели, обученной на небольшом наборе из 300 вопросов RuBQ. Используется тот факт, что мультиязычный BERT, обученный на большом количестве данных на одном языке для определенной задачи, способен показывать высокую точность при тестировании на этой же задаче на другом языке (перенос с одного языка на другой) [9,10]. В данной работе происходит перенос BERT с английского языка на русский для задач извлечения сущностей из вопросов и ранжирования отношений. Мультиязычный BERT обучается на большом количестве данных из датасета LC-QUAD2.0 [11] (датасет вопросов и соответствующих SPARQL-запросов на английском языке, около 30000 вопросов в тренировочном наборе), а затем на малом количестве вопросов на русском языке из RuBQ.

1. Краткий обзор существующих подходов, которые применяются в вопросно-ответных системах по базам знаний

Первые KBQA-системы были предназначены для ответа на простые вопросы (требующие извлечения одного триплета из базы знаний) из датасета Simple Questions [4]. Они были основаны на ранжировании триплетов из базы знаний с помощью нейронных сетей с памятью [4] или ранжировании по отдельности сущностей и отношений [12] (в качестве ответа принимался объект из триплета, для которого было максимальное скалярное произведение между векторным представлением отношения и векторным преставлением вопроса). В работах [4,12] в качестве векторного представления вопроса, сущности и отношения использовались вектора скрытого состояния BiGRU.

В [13] ответ на простые вопросы состоит из следующих этапов: извлечение сущностей из вопроса, связывание с сущностями в базе знаний, классификация вопроса по типу отношения и нахождения итогового ответа. Для извлечения сущностей и отношений использовались простые рекуррентные сети. Качество ответов на датасете Simple Questions было улучшено за счет применения BiLSTM и BiGRU.

Для ответа на сложные вопросы в работе [15] происходила генерация возможных путей в графе знаний, начинающихся с вершины-сущности, извлеченной из вопроса, и затем ранжирование путей (ответ выбирался в пути с максимальным скалярным произведением векторных представлений вопроса и сущностей; векторное представление было получено с помощью BiLSTM). В работе [16] векторные представления вопроса и пути в графе были получены с помощью TreeLSTM (учитывалась синтаксическая структура вопроса, так как на вход TreeLSTM подавалось синтаксическое дерево предложения). В системе [5] сущностям и отношениям в пути в графе присваивались начальные значения вероятностей (которые были получены от моделей извлечения сущностей и отношений). Вершины с сущностями-возможными ответами получали в качестве вероятностей средние значения вероятностей соседних вершин, В качестве ответа выбиралась вершина-ответ с максимальным значением вероятности.

В работе [17] синтаксический парсер был применен для определения шаблона SPARQL-запроса для предложений на английском языке. Аналогичный подход был использован для русского языка.

В данной работе, ввиду малого количества данных на русском языке, для обучения BERT задачам извлечения сущностей и ранжирования отношений использовался мультиязычный

BERT, обученный на большом количестве английских данных. В работе [9] исследовался перенос BERT с английского языка на русский и китайский для задачи ответа на вопросы по тексту (SQuAD). Было показано, что несмотря на то, что мультиязычный BERT, обученный на английском SQuAD, при тестировании на русском SberQuAD показывает меньшее значение F1, чем RuBERT, обученный на SberQuAD. Мультиязычный BERT способен решать задачу ответа на вопросы на русском языке, обучаясь при этом на английском языке. В статье [10] демонстрируется возможность переноса между английским, немецким, испанским и голландским языках в задаче извлечения именованных сущностей из датасета CoNLL.

2. Компоненты вопросно-ответной системы

KBQA-система состоит из следующих компонентов: построение шаблона SPARQL-запроса, извлечение сущностей из вопроса, связывание сущностей с базой знаний, ранжирование отношений.

Рассмотрим работу KBQA-системы на примере вопроса «Какой документ, подписанный 7 февраля 1992 года, положил начало Европейскому Союзу?».

Генерация SPARQL-запроса для ответа на вопрос происходит в несколько этапов:

1. Построение синтаксического дерева предложения, которое используется для определения шаблона SPARQL-запроса:

SELECT ?ans WHERE {wd:E1 p:P1 ?s . ?s ps:P2 ?ans . ?s pq:P2 DATE}.

2. Заполнение этого шаблона конкретными значениями id сущностей и отношений в Wikidata:

• извлечение подстроки, соответствующей сущности, с помощью BERT: «Европейскому Союзу»

• нахождение по этой подстроке id возможных сущностей в Wikidata: Q458, Q1377074, …

• определение с помощью BERT по описаниям сущностей, какие сущности подходят для SPARQL-запроса: Q458, Q1377074, …

• для этих сущностей в Wikidata извлечение всех возможных отношений, и далее с помощью BERT определение, какие отношения подходят для SPARQL-запроса: P457, P1343, …

• извлечение с помощью регулярных выражений численных значений или значений дат: 1992-02-07

3. Определение с помощью BERT, какие комбинации отношений подходит для SPARQL-запроса: (P457, P585), …

В результате выполнения SPARQL-запроса получается ответ на вопрос: Q11146 («Маастрихтский договор»).

3. Определение шаблона SPARQL-запроса

Для построения синтаксического дерева используется синтаксический парсер из библиотеки DeepPavlov на основе BERT (http://docs.deeppavlov.ai/en master/features/models/syntaxparser.html). Далее компонент TreeToSparql определяет шаблон SPARQL-запроса, соответствующего синтаксическому дереву.

Построение шаблона SPARQL-запроса состоит из следующих этапов:

- определение вершин синтаксического дерева, определяющих тип сущности-ответа;
- нахождение вершин, уточняющих ответ;
- нахождение вершин сущностей из вопроса;
- определение типа шаблона SPARQL-запроса.

3.1. Нахождение вершин синтаксического дерева, определяющих тип сущности-ответа

В Wikidata существует отношение “instance of” («представляет собой», «принадлежит классу»). Это отношение указывает тип сущности (например, *Москва, принадлежит классу, город*).

Вершина дерева, связанная с вопросительным местоимением, указывает на тип сущности-ответа. Например, в вопросе «Какой стране принадлежит знаменитый остров Пасхи?» с местоимением «какой» связана вершина «стране» (Рис. 1). Значит, ответом на вопрос является сущность *ent*, для которой в Wikidata есть триплет (*ent, принадлежит классу, страна*).

3.2. Нахождение вершин синтаксического дерева, уточняющих ответ

В вопросе может быть несколько возможных сущностей-ответов. При этом в вопросе

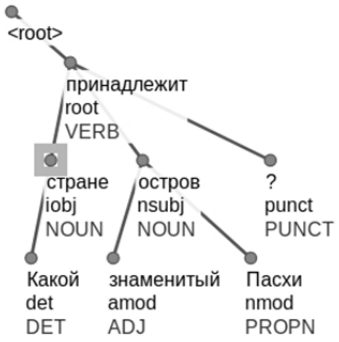

Рис. 1. Определение типа сущности-ответа по синтаксическому дереву

может содержаться уточняющая информация, которая позволяет выбрать один ответ из множества возможных. Для уточнения в вопросе может содержаться сущность, для которой в базе знаний имеется триплет с искомой сущностью. Например, в вопросе «Какой советский космический корабль в 1975 году пристыковался к американскому "Аполлону"?» вершина синтаксического дерева «советский» используется для выбора в качестве ответа именно советского космического корабля. Данный вопрос соответствует SPARQL-запросу:
SELECT ?answer WHERE { wd:Q208759 wdt:P1876 ?answer . ?answer wdt:P17 wd:Q15180 }, где Q15180 — id в Wikidata для сущности «СССР», P17 — отношение «страна».

Для нахождения уточняющих вершин в синтаксическом дереве после определения вершины, определяющей тип сущности-ответа («корабль» в данном случае) производится поиск вершин-определений, которые зависят от существительного «корабль» («космический» и «советский»).

3.3. Нахождение вершин сущностей из вопроса

На вход компонента, который строит шаблон SPARQL-запроса по синтаксическому дереву, поступает последовательность токенов вопроса и соответствующих тегов *{E-TAG, T-TAG, O-TAG}*, где *E-TAG* соответствует сущностям. Если набор токенов в соседних вершинах синтаксического дерева имеет тег *E-TAG*, эти вершины определяются как “grounded entity” («Аполлону» на Рис. 2), то есть сущность-вершина в графе знаний, от которой при выполнении SPARQL-запроса будет производиться поиск пути до сущности-ответа.

3.4. Определение типа шаблона SPARQL-запроса

Определение типа шаблона SPARQL-запроса происходит с использованием информации, извлеченной на предыдущих трех этапах: вершины сущности-ответа, вершин, уточняющих ответ, и вершин сущностей из вопроса. SPARQL-запрос должен содержать слоты для всех этих сущностей, которые затем будут заполнены конкретными id сущностей из Wikidata. Например, для вопроса из Рис. 2 была найдена одна «grounded entity» («Аполлону») и две сущности, уточняющие ответ («космический» и «советский»). Поэтому шаблон SPARQL-запроса в данном случае будет следующий:

SELECT ?ans WHERE {wd:E1 wdt:P1 ?ans . ?ans wdt:P2 wd:E2 . ?ans wdt:P3 wd:E3},

где *E1* соответствует сущности «Аполлону», *E2* и *E3* - «космический» и «советский», *ans* — искомая сущность-ответ.

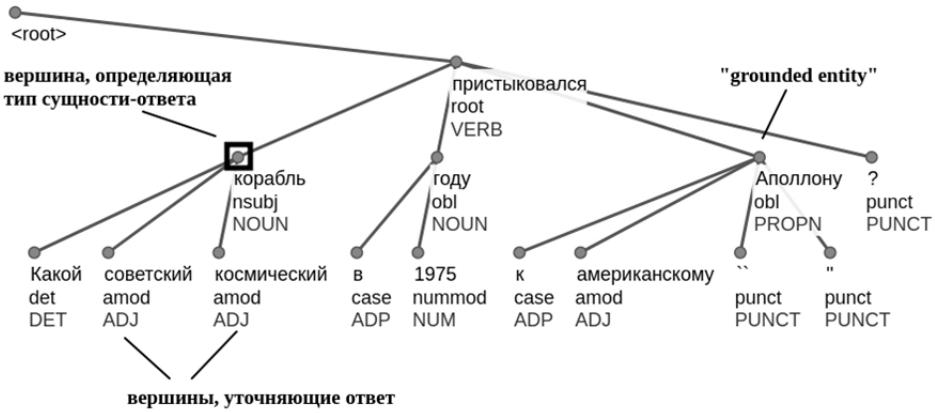

Рис. 2. Определение вершин, уточняющих ответ по синтаксическому дереву

4. Обучение мультиязычного BERT задачам извлечения сущностей и ранжирования отношений

Задача извлечения сущностей — это маркировка последовательности токенов вопроса $q = \{w_1, ..., w_n\}$ последовательностью тегов из множества *{E-TAG, T-TAG, O-TAG}*, где *E-TAG* соответствует токенам сущностей, *T-TAG* — токенам типов сущностей, *O-TAG* — остальным токенам. Например, последовательность токенов вопроса

[«Какой», «стиль», «сменила», «эпоха», «Возрождения», «?»]

соответствует последовательности тегов

["O-TAG", "T-TAG", "O-TAG", "E-TAG", "E-TAG", "O-TAG"]

(стиль Q1792644 в Wikidata) — тип сущности-ответа, эпоха Возрождения (Q4692 в Wikidata) — сущность, SPARQL-запрос для данного вопроса: SELECT ?answer WHERE {wd:Q4692 wdt:P155 ?answer . ?answer wdt:P31 wd:Q1792644}).

Вектора токенов на выходе BERT подаются на вход полносвязного слоя для классификации токенов на 3 класса (*E-TAG, T-TAG, O-TAG*).

Задача ранжирования отношений — это присвоение каждому возможному отношению $\{r_1, ..., r_m\}$ вероятностей $\{p_1, ..., p_m\}$ нахождения в SPARQL-запросе для данного вопроса.

Для ранжирования отношений на вход BERT подается последовательность токенов вопроса $q = \{w_1, ..., w_n\}$, SEP-токен, затем последовательность токенов $\{w_{r1}, ..., w_{rm}\}$ отношения r. Вектор на выходе BERT для CLS-токена подается на вход полносвязного слоя для классификации на 2 класса: 1– если r является отношением из SPARQL-запроса для вопроса q и 0 — если не является. При этом на выходе полносвязного слоя вероятность p принадлежности отношения r классу 1 используется для ранжирования возможных отношений.

В русскоязычном датасете RuBQ в тренировочном наборе содержится 300 вопросов, что недостаточно для обучения BERT задачам извлечения сущностей и ранжирования отношений. В англоязычном датасете LC-QUAD2.0, напротив, содержится более 30K вопросов. Мультиязычный BERT способен обучаться на различных языках. Поэтому был использован датасет, составленный из англоязычного LC-QUAD2.0 (93% вопросов) и русскоязычного RuBQ (7% вопросов). При этом метрики обученного BERT измерялись на тестовом наборе датасета RuBQ, чтобы определить, какое качество выделения сущностей и ранжирования отношений на русском языке способен показать мультиязычный BERT, обученный на англоязычном датасете.

Табл. 1. Ранжирование отношений, F1

Датасет для обучения BERT	F1 на тестовом наборе
LC-QUAD2.0	88,0
LC-QUAD2.0 + RuBQ	95,1

Табл. 2. Извлечение сущностей, F1

Датасет для обучения BERT	F1 на тестовом наборе
LC-QUAD2.0	63,5
LC-QUAD2.0 + RuBQ	79,1

На основании Табл. 1 и 2 можно сделать следующие выводы:

- при тестировании на русских вопросах BERT, обученный на английских вопросах, показывает качество, сравнимое с тем, которое показал BERT при тестировании на английских вопросах (F1=87 на тестовом наборе LC-QUAD2.0 в задаче извлечения сущностей и F1=89 в задаче ранжирования отношений) [18];
- качество выделения сущностей и ранжирования отношений улучшается, если в англоязычный датасет добавить небольшое количество русскоязычных примеров (в наших экспериментах 7%);
- обучение на смешанном русско-английском датасете дает приемлемое качество моделей, но при этом обучение на большом (более 10000 обучающих примеров) русскоязычном датасете (сбор которого требует значительных временных и финансовых затрат) может привести к улучшению метрик.

5. Связывание сущностей

После извлечения из текста подстрок, соответствующих сущностям, нужно найти id этих сущностей в Wikidata (связать подстроки с сущностями в Wikidata). Для этой задачи используется обратный индекс по униграммам (словарь, где ключи — это слова (униграммы), а значения — списки сущностей, которые в своем названии содержат данную униграмму). При поиске id сущности вычисляется расстояние по Левенштейну между извлеченной из

текста подстрокой и названием возможной сущности. Например, токены «эпоха» и «Возрождения» из подстроки «эпоха Возрождения» используются в качестве ключей для получения списков возможных сущностей, и сущность Q4692 с названием «эпоха Возрождения» получает максимальный score 100.

Одной и той же подстроке могут соответствовать несколько сущностей, названия которых имеют одинаковые расстояния по Левенштейну с подстрокой. Например, в вопросе «Кто написал роман "Хижина дяди Тома"?» подстроке «Хижина дяди Тома» соответствуют сущности Q2222 (литературное произведение «Хижина дяди Тома») и Q820452 (фильм «Хижина дяди Тома»). SPARQL-запрос для данного вопроса следующий: SELECT ?answer WHERE {wd:Q2222 wdt:P50 ?answer}, то есть в шаблон SPARQL-запроса должна быть подставлена сущность Q2222. У каждой сущности в Wikidata есть описание (краткое предложение, которое характеризует сущность, например, «роман Гарриет Бичер-Стоу» для Q2222 и «фильм» для Q820452), используя которое, можно определить, какая сущность больше подходит для данного SPARQL-запроса (ранжировать сущности).

Для ранжирования сущностей по контексту (т. е. вопросу) и описанию d на вход BERT подается последовательность токенов вопроса $q = \{w_1, ..., w_n\}$, SEP-токен, затем последовательность токенов $\{w_{d1}, ..., w_{dn}\}$ описания d сущности e. Вектор на выходе BERT для CLS-токена подается на вход полносвязного слоя для классификации на 2 класса: 1 – если e является сущностью из SPARQL-запроса для вопроса q и 0 — если не является. При этом на выходе полносвязного слоя вероятность p принадлежности сущности e классу 1 используется для ранжирования возможных сущностей.

6. Обзор датасета RuBQ

Датасет RuBQ содержит вопросы на русском языке и соответствующие SPARQL-запросы к Wikidata. В dev-сете RuBQ 300 вопросов, в test-сете –1200. В RuBQ вопросы подразделяются на различные типы в зависимости от количества сущностей и наличия дополнительных операций (подсчет количества сущностей, сортировка) в SPARQL-запросе (Табл. 3).

7. Результаты вопросно-ответной системы на RuBQ

В данной работе предложенная вопросно-ответная система сравнивалась с WQAqua и baseline авторов датасета RuBQ. В качестве метрики использовалась точность (accuracy) – процент правильных ответов. Ответ считался правильным, если сущность или численное значение на выходе KBQA-системы совпадало с ground-truth значением в датасете.

KBQA-система WQAqua состоит из следующих компонентов:

1. Связывание n-грамм из вопроса с сущностями из базы знаний.
2. Генерация возможных SPARQL-запросов с использованием сущностей, извлеченных на предыдущем этапе.

Табл. 3. Типы вопросов в RuBQ

Тип вопроса	Описание и пример SPARQL-запроса
1-hop	SPARQL-запрос с одним триплетом SELECT ?ans WHERE {E1 P1 ?ans}
multi-constraint	сущность-ответ входит в несколько триплетов SELECT ?ans WHERE {E1 P1 ?ans . ?ans P2 E2}
qualifier-constraint	SPARQL-запрос содержит уточняющее отношение SELECT ?ans WHERE {E1 P1 [ps:P1 ?ans; pq:P2 2004] .}
Count	используется для подсчета количества сущностей SELECT (COUNT(?ent) as ?ans) WHERE {E1 P1 ?ent}
Ranking	сущности-ответы сортируются по убыванию или возрастанию с помощью оператора ORDER SELECT ?ans WHERE { E1 P1 [ps:P1 ?ans; pq:P2 ?year] } ORDER BY ASC(?year)

Табл. 4. Точность ответов KBQA на различные типы вопросов из RuBQ

Тип вопроса	WQAqua [19],%	Baseline авторов RuBQ [8], %	Предложенная KBQA-система,%
1-hop	18.6	35.7	59.1
multi-constraint	11.1	9.9	28.3
qualifier-constraint	0	0	53.8
count, 1-hop	66	0	66
Ranking	0	0	25

3. Ранжирование возможных SPARQL-запросов по следующим параметрам:

- число слов в вопросе, которые пересекаются с названиями сущностей из SPARQL-запроса;
- расстояние по Левенштейну между n-граммами в вопросе и названиями сущностей;
- сумма числа ребер в графе знаний у сущностей из SPARQL-запроса;
- число триплетов в SPARQL-запросе.

4. Оценка confidence для SPARQL-запросов с помощью логистической регрессии.

Baseline авторов RuBQ состоит из следующих компонентов: связывание сущностей из вопроса, извлеченных с помощью синтаксического парсера, с базой знаний; нахождение отношений с помощью регулярных выражений; генерация SPARQL-запросов с использованием извлеченных сущностей и отношений.

Предложенная KBQA-система имеет следующие преимущества по сравнению с WQAqua и Baseline авторов RuBQ:

1. Определение типа SPARQL-запроса с помощью синтаксического парсера улучшает точность ответов на вопросы типа multi-constraint, qualifier-constraint и ranking. SPARQL-запрос для данных типов вопросов состоит из более чем одного триплета. Учет синтаксической структуры предложения при построении SPARQL-запроса позволяет более точно выбрать правильный тип SPARQL-запроса из множества возможных.

2. В предложенной KBQA-системе связывание сущностей и отношений с базой знаний происходит не только на основе совпадения n-грамм в тексте и названий сущностей, но и с учетом контекста (Разделы 4 и 5).

Вопросно-ответная система, рассмотренная в данной статье, показывает точность ответов на вопросы, значительно превосходящую предыдущие подходы (Табл. 4). Тем не менее, для таких типов вопросов, как multi-constraint, ranking KBQA-система дает невысокое качество (так как для типов вопросов simple question right и simple question left, аналогичных multi-constraint, в LC-QUAD2.0 была получена accuracy 67%, для типа, аналогичного rank – accuracy 48% [18]), что приводит к необходимости дальнейшего улучшения системы.

Заключение

В статье была описана вопросно-ответная система по базе знаний Wikidata. Было показано, что мультиязычный BERT способен обучаться подзадачам KBQA, таким как извлечение сущностей из вопроса и ранжирование отношений, на малом количестве данных на русском языке, что избавляет от необходимости сбора большого датасета для KBQA на русском языке. Использование синтаксического парсера помогает более точно определять тип SPARQL-запроса. Описываемая вопросно-ответная система показывает на датасете RuBQ точность ответов на вопросы, значительно превосходящую предыдущие подходы, за счет учета синтаксической информации при построении SPARQL-запроса и контекста при связывании сущностей и отношений с базой знаний.

Литература

1. Rajpurkar P., Zhang J., Lopyrev K.,. Liang P. SQuAD: 100,000+ Questions for Machine Comprehension of Text //arXiv preprint arXiv: 1606.05250. 2016.
2. Chen D., Fisch A., Weston J., Bordes A. Reading Wikipedia to answer open-domain questions //Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017. Vol.1. P.1870–1879.

3. Seo M., Lee J., Kwiatkowski T., Parikh A. P., Farhadi A., Hajishirzi H.. Real-time open-domain question answering with dense-sparse phrase index //Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2020. P. 4430–4441.
4. Bordes A., Usunier N., Chopra S.,. Weston J. Large-scale Simple Question Answering with Memory Networks //arXiv preprint arXiv: 1506.02075. 2015.
5. Vakulenko S., Garcia J. D. F., Polleres A., De Rijke M., Cochez M.. Message passing for complex question answering over knowledge graphs //Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019. P. 1431–1440.
6. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding //arXiv preprint arXiv: 1810.04805. 2018.
7. Li X., Sun X., Meng Y., Liang J., Wu F., Li J. Dice Loss for Data-imbalanced NLP Tasks //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. – 2020. P. 465–476.
8. Korablinov V., Braslavski P. RuBQ: A Russian Dataset for Question Answering over Wikidata //arXiv preprint arXiv: 2005.10659. 2020.
9. Konovalov V. P., Gulyaev P. A., Sorokin A. A., Kuratov Y. M., Burtsev M. S. Exploring the bert cross-lingual transfer for reading comprehension //Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. 2020. Vol. 2020-June. No. 19. P.445–453.
10. Pires T., Schlinger E., Garrette D. How multilingual is multilingual BERT? //Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2020. P. 4996–5001.
11. Dubey M., Banerjee D., Abdelkawi A., Lehmann J.. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia //International Semantic Web Conference. 2019. P. 69–78.
12. Dai Z., Li L., Xu W. CFO: Conditional Focused neural question answering with large-scale knowledge bases //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. Vol. 2. No. 1. P. 800–810.
13. Ture F., Jojic O. No need to pay attention: Simple recurrent neural networks work! (for answering 'simple' questions) //arXiv preprint arXiv: 1606.05029. 2017.
14. Mohammed S., Shi P., Lin J. Strong baselines for simple question answering over knowledge graphs with and without neural networks //Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018. Vol. 2. No. 2015. P. 291–296.
15. Maheshwari G. et al. Learning to rank query graphs for complex question answering over knowledge graphs //International semantic web conference. Springer. Cham. 2019. P. 487-504.
16. Zafar H., Napolitano G., Lehmann J. Formal Query Generation for Question Answering over Knowledge Bases //European Semantic Web Conference. 2018. P. 714–728.
17. Ochieng P. PAROT: Translating natural language to SPARQL //Expert Systems with Applications: X. 2020. Vol. 5. P. 100024.
18. Evseev D. A., Arkhipov M. Y. Sparql query generation for complex question answering with bert and bilstm-based model //Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. 2020. Vol. 2020-June. No. 19. P. 270–282.
19. Dieenbach D. et al. Towards a question answering system over the semantic web //Semantic Web. 2020. Vol. 11. No. 3. P. 421-439.

Евсеев Дмитрий Андреевич. Младший инженер-исследователь лаборатории нейронных систем и глубокого обучения. Московский физико-технический институт. Область научных интересов: обработка естественного языка, глубокое обучение. E-mail: dmitrij.euseew@yandex.ru

Query Generation for Complex Question Answering in Russian with Syntax Parser

D. A. Evseev

Moscow Institute of Physics and Technology, Moscow, Russia

Abstract. This paper describes the system which translates a natural language question into a SPARQL-query. The question answering system consists of: the syntax parser, which builds a syntax tree of a sentence; the component, which defines the SPARQL query template using the syntax tree; models, which find entities and relations to fill in the slots of the SPARQL query template. We use BERT for entity detection and relation ranking. One of the characteristics of BERT training on knowledge base question answering subtasks in Russian is small amount of training data. Due to this, we investigate training of multilingual BERT, pretrained on LC-QUAD2.0 dataset, on entity detection and relation ranking tasks on small amount of Russian samples from RuBQ dataset. The proposed question answering system outperforms previous approaches on RuBQ dataset.

Keywords: question answering system, knowledge base, query generation, multilingual BERT.

DOI 10.14357/20718594210305

References

1. Rajpurkar P., Zhang J., Lopyrev K., Liang P. SQuAD: 100,000+ Questions for Machine Comprehension of Text //arXiv preprint arXiv: 1606.05250. 2016.
2. Chen D., Fisch A., Weston J., Bordes A. Reading Wikipedia to answer open-domain questions //Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017. Vol. 1. P. 1870–1879.
3. Seo M., Lee J., Kwiatkowski T., Parikh A., Farhadi A., Hajishirzi H. Real-time open-domain question answering with dense-sparse phrase index //Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2020. P.4430–4441.
4. Bordes A., Usunier N., Chopra S., Weston J. Large-scale Simple Question Answering with Memory Networks //arXiv preprint arXiv: 1506.02075. 2015.
5. Vakulenko S., Garcia J. D. F., Polleres A., De Rijke M., Cochez M. Message passing for complex question answering over knowledge graphs //Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019. P. 1431–1440.
6. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding //arXiv preprint arXiv: 1810.04805. 2018.
7. Li X., Sun X., Meng Y., Liang J., Wu F., Li J. Dice Loss for Data-imbalanced NLP Tasks //Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020. P. 465–476.
8. Korablinov V., Braslavski P. RuBQ: A Russian Dataset for Question Answering over Wikidata //arXiv preprint arXiv: 2005.10659. 2020.
9. Konovalov V. P., Gulyaev P. A., Sorokin A. A., Kuratov Y. M., Burtsev M. S. Exploring the bert cross-lingual transfer for reading comprehension //Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. 2020. Vol. 2020-June. No.19. P.445–453.
10. Pires T., Schlinger E., Garrette D. How multilingual is multilingual BERT? //Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2020. P. 4996–5001.
11. Dubey M., Banerjee D., Abdelkawi A., Lehmann J. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia //International Semantic Web Conference. 2019. P. 69–78.
12. Dai Z., Li L., Xu W. CFO: Conditional Focused neural question answering with large-scale knowledge bases //Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016. Vol. 2. No. 1. P. 800–810.
13. Ture F., Jojic O. No need to pay attention: Simple recurrent neural networks work! (for answering 'simple' questions) //arXiv preprint arXiv: 1606.05029. 2017.
14. Mohammed S., Shi P., Lin J. Strong baselines for simple question answering over knowledge graphs with and without neural networks //Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018. Vol.2. No. 2015. P. 291–296.
15. Maheshwari G. et al. Learning to rank query graphs for complex question answering over knowledge graphs //International semantic web conference. Springer. Cham. 2019. P. 487-504.
16. Zafar H., Napolitano G., Lehmann J. Formal Query Generation for Question Answering over Knowledge Bases //European Semantic Web Conference. 2018. P.714–728.
17. Ochieng P. PAROT: Translating natural language to SPARQL //Expert Systems with Applications: X. 2020. Vol. 5. P. 100024.
18. Evseev D. A., Arkhipov M. Y. Sparql query generation for complex question answering with bert and bilstm-based model //Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. 2020. Vol. 2020-June. No. 19. P. 270–282..
19. Diefenbach D. et al. Towards a question answering system over the semantic web //Semantic Web. 2020. Vol. 11. No. 3. P. 421-439.

Evseev Dmitry A. Junior engineer-researcher, Neural Networks and Deep Learning Laboratory, Moscow Institute of Physics and Technology. Research areas: natural language processing, deep learning. E-mail: dmitrij.euseew@yandex.ru