RuBQ: A Russian Dataset for Question Answering over Wikidata

Vladislav Korablinov[1]* and Pavel Braslavski[2,3][0000−0002−6964−458X]

[1] ITMO University, Saint Petersburg, Russia
vladislav.korablinov@gmail.com
[2] JetBrains Research, Saint Petersburg, Russia
[3] Ural Federal University, Yekaterinburg, Russia
pbras@yandex.ru

Abstract. The paper presents **RuBQ**, the first Russian knowledge base question answering (KBQA) dataset. The high-quality dataset consists of 1,500 Russian questions of varying complexity, their English machine translations, SPARQL queries to Wikidata, reference answers, as well as a Wikidata sample of triples containing entities with Russian labels. The dataset creation started with a large collection of question-answer pairs from online quizzes. The data underwent automatic filtering, crowd-assisted entity linking, automatic generation of SPARQL queries, and their subsequent in-house verification. The freely available dataset will be of interest for a wide community of researchers and practitioners in the areas of Semantic Web, NLP, and IR, especially for those working on multilingual question answering. The proposed dataset generation pipeline proved to be efficient and can be employed in other data annotation projects.

Keywords: Knowledge base question answering · Semantic parsing · Evaluation · Russian language resources
Resource location: http://doi.org/10.5281/zenodo.3835913
Project page: https://github.com/vladislavneon/RuBQ

1 Introduction

Question answering (QA) addresses the task of returning a precise and concise answer to a natural language question posed by the user. QA received a great deal of attention both in academia and industry. Two main directions within QA are *Open-Domain Question Answering (ODQA)* and *Knowledge Base Question Answering (KBQA)*. ODQA searches for the answer in a large collection of text documents; the process is often divided into two stages: 1) retrieval of potentially relevant paragraphs and 2) spotting an answer span within the paragraph (referred to as *machine reading comprehension, MRC*). In contrast, KBQA uses a *knowledge base* as a source of answers. A knowledge base is a large collection

* Work done as an intern at JetBrains Research.

of factual knowledge, commonly structured in subject–predicate–object (SPO) triples, for example `(Vladimir_Nabokov, spouse, Véra_Nabokov)`.

A potential benefit of KBQA is that it uses knowledge in a distilled and structured form that enables reasoning over facts. In addition, knowledge base structure is inherently language-independent – entities and predicates are assigned unique identifiers that are tied to specific languages through labels and descriptions, – which makes KBs more suitable for multilingual QA. The task of KBQA can be formulated as a translation from natural language question into a formal KB query (expressed in SPARQL, SQL, or λ-calculus). In many real-life applications, like in *Jeopardy!* winning IBM Watson [15] and major search engines, hybrid QA systems are employed – they rely on both text document collections and structured knowledge bases.

High-quality annotated data is crucial for measurable progress in question answering. Since the advent of SQuAD [27], a wide variety of datasets for machine reading comprehension have emerged, see a recent survey [39]. We are witnessing a growing interest in multilingual question answering, which leads to the creation of multilingual MRC datasets [24,1,8]. Multilingual KBQA has received a deal of attention in the literature [16,9]. However, almost all available KBQA datasets are English, Chinese datasets being an exception. Existing multilingual QALD datasets are rather small.

In this paper we present **RuBQ** (pronounced [‘rubik]) – **Ru**ssian Knowledge **B**ase **Q**uestions, a KBQA dataset that consists of 1,500 Russian questions of varying complexity along with their English machine translations, corresponding SPARQL queries, answers, as well as a subset of Wikidata covering entities with Russian labels. To the best of our knowledge, this is the first Russian KBQA and semantic parsing dataset. To construct the dataset, we started with a large collection of trivia Q&A pairs harvested on the Web. We built a dedicated recall-oriented Wikidata entity linking tool and verified the obtained answers’ candidate entities via crowdsourcing. Then, we generated paths between possible question entities and answer entities and carefully verified them.

The freely available dataset is of interest for a wide community of Semantic Web, natural language processing (NLP), and information retrieval (IR) researchers and practitioners, who deal with multilingual question answering. The proposed dataset generation pipeline proved to be efficient and can be employed in other data annotation projects.

2 Related work

Table 1 summarizes the characteristics of KBQA datasets that have been developed to date. These datasets vary in size, underlying knowledge base, presence of questions’ logical forms and their formalism, question types and sources, as well as the language of the questions.

The questions of the earliest Free917 dataset [7] were generated by two people without consulting a knowledge base, the only requirement was a diversity of questions’ topics; each question is provided with its logical form to query

Table 1. KBQA datasets. Target knowledge base (**KB**): Fb – Freebase, DBp – DBpedia, Wd – Wikidata (MSParS description does not reveal the details about the KB associated with the dataset). **CQ** indicates the presence of complex questions in the dataset. Logical form (**LF**) annotations: λ – lambda calculus, S – SPARQL queries, t – SPO triples. Question generation method (**QM**): M – manual generation from scratch, SE – search engine query suggest API, L – logs, T+PP – automatic generation of question surrogates based on templates followed by crowdsourced paraphrasing, CS – crowdsourced manual generation based on formal representations, QZ – quiz collections, FA – fully automatic generation based on templates.

Dataset	**Year**	**#Q**	**KB**	**CQ**	**LF**	**QM**	**Lang**
Free917 [7]	2013	917	Fb	+	λ	M	en
WebQuestions [3]	2013	5,810	Fb	+	–	SE	en
SimpleQuestions [5]	2015	108,442	Fb	–	t	CS	en
ComplexQuestions [2]	2016	2,100	Fb	+	–	L, SE	en
GraphQuestions [30]	2016	5,166	Fb	+	S	T+PP	en
WebQuestionsSP [38]	2016	4,737	Fb	+	S	SE	en
SimpleQuestions2Wikidata [11]	2017	21,957	Wd	–	t	CS	en
30M Factoid QA Corpus [29]	2017	30M	Fb	–	t	FA	en
LC QuAD [32]	2017	5,000	DBp	+	S	T+PP	en
ComplexWebQuestions [31]	2018	34,689	Fb	+	S	T+PP	en
ComplexSequentialQuestions [28]	2018	1.6M	Wd	+	–	M+CS+FA	en
QALD9 [33]	2018	558	DBp	+	S	L	mult
LC-QuAD 2.0 [13]	2019	30,000	DBp, Wd	+	S	T+PP	en
FreebaseQA[19]	2019	28,348	Fb	+	S	QZ	en
MSParS [12]	2019	81,826	–	+	λ	T+PP	zh
CFQ [21]	2020	239,357	Fb	+	S	FA	en
RuBQ (this work)	2020	1,500	Wd	+	S	QZ	ru

Freebase. Berant et al. [3] created WebQuestions dataset that is significantly larger but does not contain questions' logical forms. Questions were collected through Google suggest API: authors fed parts of the initial question to the API and repeated the process with the returned questions until 1M questions were reached. After that, 100K randomly sampled questions were presented to MTurk workers, whose task was to find an answer entity in Freebase. Later studies have shown that only two-thirds of the questions in the dataset are completely correct; many questions are ungrammatical and ill-formed [38,37]. Yih et al. [38] enriched 81.5% of WebQuestions with SPARQL queries and demonstrated that semantic parses substantially improve the quality of KBQA. They also showed that semantic parses can be obtained at an acceptable cost when the task is broken down into smaller steps and facilitated by a handy interface. Annotation was performed by five people familiar with Freebase design, which hints at the fact that the task is still too tough for crowdsourcing. WebQuestions were used in further studies aimed to generate complex questions [2,31].

SimpleQuestions [5] is the largest manually created KBQA dataset to date. Instead of providing logical parses for existing questions, the approach explores the opposite direction: based on formal representation, a natural language question is generated by crowd workers. First, the authors sampled SPO triples from a Freebase subset, favoring non-frequent subject–predicate pairs. Then, the triples were presented to crowd workers, whose task was to generate a question about the subject, with the object being the answer. This approach doesn't guarantee that the answer is unique – Wu et al. [37] estimate that SOTA results on the dataset (about 80% correct answers) reach its upper bound, since the rest of the questions are ambiguous and cannot be answered precisely. The dataset was used for the fully automatic generation of a large collection of natural language questions from Freebase triples with neural machine translation methods [29]. Dieffenbach et al. [11] succeeded in a semi-automatic matching of about one-fifth of the dataset to Wikidata.

The approach behind FreebaseQA dataset [19] is the closest to our study – it builds upon a large collection of trivia questions and answers (borrowed largely from TriviaQA dataset for reading comprehension [20]). Starting with about 130K Q&A pairs, the authors run NER over questions and answers, match extracted entities against Freebase, and generate paths between entities. Then, human annotators verify automatically generated paths, which resulted in about 28K items marked relevant. Manual probing reveals that many questions' formal representations in the dataset are not quite precise. For example, the question `eval-25`: *Who captained the Nautilus in 20,000 Leagues Under The Sea?* is matched with the relation *book.book.characters* that doesn't represent its meaning and leads to multiple answers along with a correct one (*Captain Nemo*). Our approach differs from the above in several aspects. We implement a recall-oriented IR-based entity linking since many questions involve general concepts that cannot be recognized by off-the-shelf NER tools. After that, we verify answer entities via crowdsourcing. Finally, we perform careful in-house verification of automatically generated paths between question and answer entities in KB. We can conclude that our pipeline leads to a more accurate representation of questions' semantics.

The questions in the KBQA datasets can be *simple*, i.e. corresponding to a single fact in the knowledge base, or *complex*. Complex questions require a combination of multiple facts to answer them. WebQuestions consists of 85% simple questions; SimpleQuestions and 30M factoid QA Corpus contain only simple questions. Many studies [13,2,12,31,28,21] purposefully target complex questions.

The majority of datasets use Freebase [4] as target knowledge base. Freebase was discontinued and exported to Wikidata [25]; the latest available Freebase dump dates back to early 2016. Three collections [33,32,13] use DBpedia [22]. Newer datsets [25,28,13] use Wikidata [36], which is much larger, up-to-date, and has more multilingual labels and descriptions. The majority of datasets, where natural language questions are paired with logical forms, employ SPARQL as a more practical and immediate option compared to lambda calculus.

Existing KBQA datasets are almost exclusively English, with Chinese MSParS dataset being an exception [12]. QALD-9 [33], the latest edition of QALD shared task,[4] contains questions in 11 languages: English, German, Russian, Hindi, Portuguese, Persian, French, Romanian, Spanish, Dutch, and Italian. The dataset is rather small; at least Russian questions appear to be non-grammatical machine translations.

There are several studies on knowledge base question generation [29,14,17,21]. These works vary in the amount and form of supervision, as well as the structure and the complexity of the generated questions. However, automatically generated questions are intended primarily for training; the need for high-quality, human-annotated data for testing still persists.

3 Dataset Creation

Following previous studies [19,20], we opted for quiz questions that can be found in abundance online along with the answers. These questions are well-formed and diverse in terms of properties and entities, difficulty, and vocabulary, although we don't control these properties directly during data processing and annotation.

The dataset generation pipeline consists of the following steps: 1) data gathering and cleaning; 2) entity linking in answers and questions; 3) verification of answer entities by crowd workers; 4) generation of paths between answer entities and question candidate entities; 5) in-house verification/editing of generated paths. In parallel, we created a Wikidata sample containing all entities with Russian labels. This snapshot mitigates the problem of Wikidata's dynamics – a reference answer may change with time as the knowledge base evolves. In addition, the smaller dataset lowers the threshold for KBQA experiments. In what follows we elaborate on these steps.

3.1 Raw Data

We mined about 150,000 Q&A pairs from several open Russian quiz collections on the Web.[5] We found out that many items in the collection aren't actual factoid questions, for example, cloze quizzes (*Leonid Zhabotinsky was a champion of Olympic games in . . . [Tokyo]*[6]), crossword, definition, and multi-choice questions, as well as puzzles (*Q: There are a green one, a blue one, a red one and an east one in the white one. What is this sentence about? A: The White House*). We compiled a list of Russian question words and phrases and automatically removed questions that don't contain any of them. We also removed duplicates and crossword questions mentioning the number of letters in the expected answer. This resulted in 14,435 Q&A pairs.

[4] See overview of previous QALD datasets in [34].

[5] `http://baza-otvetov.ru`, `http://viquiz.ru`, and others.

[6] Hereafter English examples are translations from original Russian questions and answers.

3.2 Entity Linking in Answers and Questions

We implemented an IR-based approach for generating Wikidata entity candidates mentioned in answers and questions. First, we collected all Wikidata entities with Russian labels and aliases. We filtered out Wikimedia disambiguation pages, dictionary and encyclopedic entries, Wikimedia categories, Wikinews articles, and Wikimedia list articles. We also removed uninformative entities with less than four outgoing relations. These steps resulted in 4,114,595 unique entities with 5,430,657 different labels and aliases.

After removing punctuation, we indexed the collection with Elasticsearch using built-in tokenization and stemming. Each text string (question or answer) produces three types of queries to the Elasticsearch index: 1) all token trigrams; 2) capitalized bigrams (many named entities follow this pattern, e.g. *Alexander Pushkin*, *Black Sea*); and 3) free text query containing only nouns, adjectives, and numerals from the original string. N-gram queries (types 1 and 2) are run as phrase queries, whereas recall-oriented free text queries (type 3) are executed as Elasticsearch fuzzy search queries. Results of the latter search are re-ranked using a combination of BM25 scores from Elasticsearch and page view statistics of corresponding Wikipedia articles. Finally, we combine search results preserving the type order and retain Top-10 results for further processing. The proposed approach effectively combines precision- (types 1 and 2) and recall-oriented (type 3) processing.

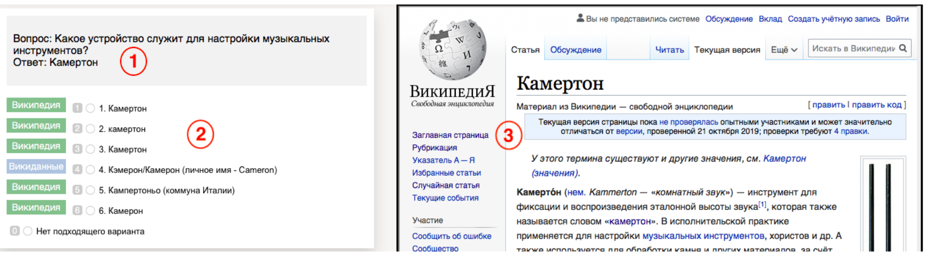
3.3 Crowdsourcing Annotations

Entity candidates for answers obtained through the entity linking described above were verified on Yandex.Toloka crowdsourcing platform.[7] Crowd workers were presented with a Q&A pair and a ranked list of candidate entities. In addition, they could consult a Wikipedia page corresponding to the Wikidata item, see Figure 1. The task was to select a single entity from the list or the *None of the above* option. The average number of candidates on the list is 5.43.

Crowd workers were provided with a detailed description of the interface and a variety of examples. To proceed to the main task, crowd workers had to first pass a qualification consisting of 20 tasks covering various cases described in the instruction. We also included 10% of honeypot tasks for live quality monitoring. These results are in turn used for calculating confidence of the annotations obtained so far as a weighted majority vote (see details in [18]). Confidence value governs overlap in annotations: if the confidence is below 0.85, the task is assigned to the next crowd worker. We hired Toloka workers from the best 30% cohort according to internal rating. As a result, the average confidence for the annotation is 98.58%; the average overlap is 2.34; average time to complete a task is 19 seconds.

In total, 9,655 out of 14,435 answers were linked to Wikidata entities. Among the matched entities, the average rank of the correct candidate appeared to be

[7] `https://toloka.yandex.com/`



Вопрос: Какое устройство служит для настройки музыкальных инструментов?
Ответ: Камертон

- Википедия 1 ○ 1. Камертон
- Википедия 2 ○ 2. камертон
- Википедия 3 ○ 3. Камертон
- Викиданные 4 ○ 4. Кэмерон/Камерон (личное имя - Cameron)
- Википедия 5 ○ 5. Кампертоньо (коммуна Италии)
- Википедия 6 ○ 6. Камерон
- 0 ○ Нет подходящего варианта

Камертон

Материал из Википедии — свободной энциклопедии

Текущая версия страницы пока не проверялась опытными участниками и может значительно отличаться от версии, проверенной 21 октября 2019; проверки требуют 4 правки.

У этого термина существуют и другие значения, см. Камертон (значения).

Камерто́н (нем. *Kammerton* — «*комнатный звук*») — инструмент для фиксации и воспроизведения эталонной высоты звука[1], которая также называется словом «камертон». В исполнительской практике применяется для настройки музыкальных инструментов, хористов и др. А также используется для обработки камня и других материалов, за счёт

Fig. 1. Interface for crowdsourced entity linking in answers: 1 – question and answer; 2 – entity candidates; 3 – Wikpedia page for a selected entity from the list of candidates (in case there is no associated Wikipedia page, the Wikidata item is shown).

1.5. The combination of automatic candidate generation and subsequent crowdsourced verification proved to be very efficient. A possible downside of the approach is a lower share of literals (dates and numerical values) in the annotated answers. We could match only a fraction of those answers with Wikidata: Wikidata's standard formatted literals may look completely different even if representing the same value. Out of 1,255 date and numerical answers, 683 were linked to a Wikidata entity such as a particular year. For instance, the answer for *In what year was Immanuel Kant born?* matches *Q6926 (year 1724)*, whereas the corresponding Wikidata value is `"1724-04-22"^^xsd:dateTime`. Although the linkage is deemed correct, this barely helps generate a correct path between question and answer entities.

3.4 Path Generation and In-house Annotation

We applied entity linking described above to the 9,655 questions with verified answers and obtained 8.56 candidate entities per question on average. Next, we generated candidate subgraphs spanning question and answer entities, restricting the length between them by two hops. We examined the questions in the sample and found out that longer distances between question and answer entities are very rare.

We investigated the option of filtering out erroneous question entities using crowdsourcing analogous to answer entity verification. A pilot experiment on a small sample of questions showed that this task is much harder – we got only 64% correct matches on a test set. Although the average number of generated paths decreased (from 1.9 to 0.9 and from 6.2 to 3.5 for paths of length one and two, respectively), it also led to losing correct paths for 14% of questions. Thus, we decided to perform an in-house verification of the generated paths. The work was performed by the authors of the paper.

After sending queries to the Wikidata endpoint, we were able to find chains of length one or two for 3,194 questions; the remaining 6,461 questions were left unmatched. We manually inspected 200 random unmatched questions and found out that only 10 of them could possibly be answered with Wikidata, but the required facts are missing in the KB.

Out of 2,809 1-hop candidates corresponding to 1,799 questions, 866 were annotated as correct. For the rest 2,328 questions, we verified 3,591 2-hop candidates, but only 55 of them were deemed correct. 279 questions were marked as answerable with Wikidata. To increase the share of complex questions in the dataset, we manually constructed SPARQL queries for them.

Finally, we added 300 questions marked as non-answerable over Wikidata, although their answers are present in the knowledge base. These adversarial examples are akin to unanswerable questions in the second edition of SQuAD dataset [26]. The majority of these questions are unanswerable because required predicates are missing in Wikidata, e.g. *How many bells does the tower of Pisa have? (7)*. In some cases, although both question and answer entities are present, the relation between them is missing, e.g. *What circus was founded by Albert Vilgelmovich Salamonsky in 1880? (Moscow Circus on Tsvetnoy Boulevard)*. The presence of such questions makes the task more challenging and realistic.

4 RuBQ Dataset

4.1 Dataset Statistics

Our dataset has 1,500 unique questions in total. It mentions 2,357 unique entities – 1,218 in questions and 1,250 in answers. There are 242 unique relations in the dataset. The average length of the original questions is 7.99 words (median 7); machine-translated English questions are 10.58 words on average (median 10). 131 questions have more than one correct answer. For 1,154 questions the answers are Wikidata entities, and for 46 questions the answers are literals.

Inspired by a taxonomy of query complexity in LC QuAD 2.0 [13], we annotated obtained SPARQL queries in a similar way. The query type is defined by the constraints in the SPARQL query, see Table 2. Note that some queries have multiple type tags. For example, SPARQL query for the question *How many moons does Mars have?* is assigned *1-hop* and *count* types and therefore isn't simple in terms of SimpleQuestions dataset.

Taking into account RuBQ's modest size, we propose to use the dataset primarily for testing rule-based systems, cross-lingual transfer learning models, and models trained on automatically generated examples, similarly to recent MRC datasets [8,1,24]. We split the dataset into development (300) and test (1,200) sets in such a way to keep a similar distribution of query types in both subsets.

4.2 Dataset Format

For each entry in the dataset, we provide: the original question in Russian, machine-translated English question obtained through Yandex.Translate,[8] original answer text (may differ textually from the answer entity's label retrieved from Wikidata), SPARQL query representing the meaning of the question, a

[8] `https://translate.yandex.com/`

Table 2. Query types in RuBQ (#D/T – number of questions in development and test subsets, respectively).

Type	#D/T	Description
1-hop	198/760	Query corresponds to a single SPO triple
multi-hop	14/55	Query's constraint is applied to more than one fact
multi-constraint	21/110	Query contains more than one SPARQL constraint
qualifier-answer	1/5	Answer is a value of a qualifier relation, similar to "fact with qualifiers" in LC-QuAD 2.0
qualifier-constraint	4/22	Query poses constraints on qualifier relations; a superclass of "temporal aspect" in LC-QuAD 2.0
reverse	6/29	Answer's variable is a subject in at least one constraint
count	1/4	Query applies `COUNT` operator to the resulting entities, same as in LC-QuAD 2.0
ranking	3/16	`ORDER` and `LIMIT` operators are applied to the entities specified by constraints, same as in LC-QuAD 2.0
0-hop	3/12	Query returns an entity already mentioned in the questions. The corresponding questions usually contain definitions or entity's alternative names
exclusion	4/18	Query contains `NOT IN`, which excludes entities mentioned in the question from the answer
no-answer	60/240	Question cannot be answered with the knowledge base, although answer entity may be present in the KB

list of entities in the query, a list of relations in the query, a list of answers (a result of querying the Wikidata subset, see below), and a list of query type tags, see Table 3 for examples. RuBQ is distributed under CC BY-SA license and is available in JSON format.

The dataset is accompanied by `RuWikidata8M` – a Wikidata sample containing all the entities with Russian labels.[9] It consists of about 212M triples with 8.1M unique entities. As mentioned before, the sample guarantees the correctness of the queries and answers and makes the experiments with the dataset much simpler. For each entity, we executed a series of `CONSTRUCT` SPARQL queries to retrieve all the truthy statements and all the full statements with their linked data.[10] We also added all the triples with `subclass of (P279)` predicate to the sample. This class hierarchy can be helpful for question answering task in the absence of an explicit ontology in Wikidata. The sample contains Russian and English labels and aliases for all its entities.

4.3 Baselines

We provide two RuBQ baselines from third-party systems – DeepPavlov and WDAqua – that illustrate two possible approaches to cross-lingual KBQA.

[9] `https://zenodo.org/record/3751761`, project's page on github points here.

[10] Details about Wikidata statement types can be found here: `https://www.mediawiki.org/wiki/Wikibase/Indexing/RDF_Dump_Format#Statement_types`

Table 3. Examples from the RuBQ dataset. Answer entities' labels are not present in the dataset and are cited here for convenience. Note that the original Q&A pair corresponding to the third example below contains only one answer – *geodesist*.

Question	Who wrote the novel "Uncle Tom's Cabin"?

SPARQL query	`SELECT ?answer`
	`WHERE {`
	`  wd:Q2222 wdt:P50 ?answer`
	`}`

Answers IDs	Q102513 (Harriet Beecher Stowe)
Tags	1-hop
Question	Who played Prince Andrei Bolkonsky in Sergei Bondarchuk's film "War and Peace"?

SPARQL query	`SELECT ?answer`
	`WHERE {`
	`  wd:Q845176 p:P161`
	`  [ ps:P161 ?answer; pq:P453 wd:Q2737140 ]`
	`}`

Answers IDs	Q312483 (Vyacheslav Tikhonov)
Tags	qualifier-constraint
Question	Who uses a theodolite for work?

SPARQL query	`SELECT ?answer`
	`WHERE {`
	`  wd:Q181517 wdt:P366 [ wdt:P3095 ?answer ]`
	`}`

Answers IDs	Q1734662 (cartographer), Q11699606 (geodesist), Q294126 (land surveyor)
Tags	multi-hop

To the best of our knowledge, the KBQA library[11] from an open NLP framework DeepPavlov [6] is the only freely available KBQA implementation for Russian language. The library uses Wikidata as a knowledge base and implements the standard question processing steps: NER, entity linking, and relation detection. According to the developers of the library, they used machine-translated SimpleQuestions and a dataset for zero-shot relation extraction [23] to train the model. The library returns a single string or *not found* as an answer. We obtained an answer entity ID using reverse ID-label mapping embedded in the model. If no ID is found, we treated the answer as a literal.

WDAqua [10] is a rule-based KBQA system that answers questions in several languages using Wikidata. WDAqua returns a (possibly empty) ranked list of Wikidata item IDs along with corresponding SPARQL queries. We obtain WDAqua's answers by sending RuBQ questions machine-translated into English to its API.[12]

[11] `http://docs.deeppavlov.ai/en/master/features/models/kbqa.html`

[12] `www.wdaqua.eu/qa`

Table 4. DeepPavlov's and WDAqua's top-1 results on RuBQ's answerable and unanswerable questions in the test set, and the breakdown of correct answers by query type.

	DeepPavlov	WDAqua
Answerable (960)		
correct	129	153
1-hop	123	136
1-hop + reverse	0	3
1-hop + count	0	2
1-hop + exclusion	0	2
multi-constraint	4	9
multi-hop	1	0
qualifier-constraint	1	0
qualifier-answer	0	1
incorrect/empty	831	807
Unanswerable (240)		
incorrect	65	138
empty/not found	175	102

WDAqua outperforms DeepPavlov in terms of precision@1 on the answerable subset (16% vs. 13%), but demonstrates a lower accuracy on unanswerable questions (43% vs. 73%). Table 4 presents detailed results. In contrast to DeepPavlov, WDAqua returns a ranked list of entities as a response to the query, and for 23 out of 131 questions with multiple correct answers, it managed to perfectly match the set of answers. For eight questions with multiple answers, WDAqua's top-ranked answers were correct, but the lower-ranked ones contained errors. To facilitate different evaluation scenarios, we provide an evaluation script that calculates precision@1, exact match, and precision/recall/F1 measures, as well as the breakdown of results by query types.

5 Conclusion and Future Work

We presented RuBQ – the first Russian dataset for Question Answering over Wikidata. The dataset consists of 1,500 questions, their machine translations into English, and annotated SPARQL queries. 300 RuBQ questions are unanswerable, which poses a new challenge for KBQA systems and makes the task more realistic. The dataset is based on a collection of quiz questions. The data generation pipeline combines automatic processing, crowdsourced and in-house verification, and proved to be very efficient. The dataset is accompanied by a Wikidata sample of 212M triples that contain 8.1M entities with Russian and English labels, and an evaluation script. The provided baselines demonstrate the feasibility of the cross-lingual approach in KBQA, but at the same time indicate there is ample room for improvements. The dataset is of interest for a wide community of researchers in the fields of Semantic Web, Question Answering, and Semantic Parsing.

In the future, we plan to explore other data sources and approaches for RuBQ expansion: search query suggest APIs as for WebQuestions [3], a large question log [35], and Wikidata SPARQL query logs.[13] We will also address complex questions and questions with literals as answers, as well as the creation of a stronger baseline for RuBQ.

Acknowledgments. We thank Mikhail Galkin, Svitlana Vakulenko, Vladimir Kovalenko, Yaroslav Golubev, and Rishiraj Saha Roy for their valuable comments and fruitful discussion on the paper draft. We also thank Pavel Bakhvalov, who helped collect `RuWikidata8M` sample and contributed to the first version of the entity linking tool. We are grateful to Yandex.Toloka for their data annotation grant.

References

1. Artetxe, M., Ruder, S., Yogatama, D.: On the cross-lingual transferability of monolingual representations. arXiv preprint arXiv:1910.11856 (2019)
2. Bao, J., Duan, N., Yan, Z., Zhou, M., Zhao, T.: Constraint-based question answering with knowledge graph. In: COLING. pp. 2503–2514 (2016)
3. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs. In: EMNLP. pp. 1533–1544 (2013)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD. pp. 1247–1250 (2008)
5. Bordes, A., Usunier, N., Chopra, S., Weston, J.: Large-scale simple question answering with memory networks. arXiv preprint arXiv:1506.02075 (2015)
6. Burtsev, M., et al.: Deeppavlov: Open-source library for dialogue systems. In: ACL (System Demonstrations). pp. 122–127 (2018)
7. Cai, Q., Yates, A.: Large-scale semantic parsing via schema matching and lexicon extension. In: ACL. pp. 423–433 (2013)
8. Clark, J.H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., Palomaki, J.: TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. arXiv preprint arXiv:2003.05002 (2020)
9. Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the semantic web. arXiv preprint arXiv:1803.00832 (2018)
10. Diefenbach, D., Singh, K.D., Maret, P.: WDAqua-core1: A question answering service for RDF knowledge bases. In: WWW Companion Volume. pp. 1087–1091 (2018)
11. Diefenbach, D., Tanon, T.P., Singh, K.D., Maret, P.: Question answering benchmarks for Wikidata. In: ISWC (Posters & Demonstrations) (2017)
12. Duan, N.: Overview of the NLPCC 2019 shared task: Open domain semantic parsing. In: NLPCC. pp. 811–817 (2019)
13. Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J.: LC-QuAD 2.0: A large dataset for complex question answering over Wikidata and DBpedia. In: ISWC (2019)

[13] `https://iccl.inf.tu-dresden.de/web/Wikidata_SPARQL_Logs/en`

14. Elsahar, H., Gravier, C., Laforest, F.: Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In: NAACL. pp. 218–228 (2018)
15. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., et al.: Building Watson: An overview of the DeepQA project. AI magazine **31**(3), 59–79 (2010)
16. Hakimov, S., Jebbara, S., Cimiano, P.: AMUSE: multilingual semantic parsing for question answering over linked data. In: ISWC. pp. 329–346 (2017)
17. Indurthi, S.R., Raghu, D., Khapra, M.M., Joshi, S.: Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In: EACL. pp. 376–385 (2017)
18. Ipeirotis, P.G., Provost, F., Sheng, V.S., Wang, J.: Repeated labeling using multiple noisy labelers. Data Mining and Knowledge Discovery **28**(2), 402–441 (2014)
19. Jiang, K., Wu, D., Jiang, H.: FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In: NAACL. pp. 318–323 (2019)
20. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In: ACL. pp. 1601–1611 (2017)
21. Keysers, D., Schärli, N., Scales, N., et al.: Measuring compositional generalization: A comprehensive method on realistic data. In: ICLR (2020)
22. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
23. Levy, O., Seo, M., Choi, E., Zettlemoyer, L.: Zero-shot relation extraction via reading comprehension. In: CoNLL. pp. 333–342 (2017)
24. Lewis, P., Oğuz, B., Rinott, R., Riedel, S., Schwenk, H.: MLQA: Evaluating cross-lingual extractive question answering. arXiv preprint arXiv:1910.07475 (2019)
25. Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L.: From Freebase to Wikidata: The great migration. In: WWW. pp. 1419–1428 (2016)
26. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for SQuAD. In: ACL. pp. 784–789 (2018)
27. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: EMNLP. pp. 2383–2392 (2016)
28. Saha, A., Pahuja, V., Khapra, M.M., Sankaranarayanan, K., Chandar, S.: Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In: AAAI (2018)
29. Serban, I.V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., Bengio, Y.: Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In: ACL. pp. 588–598 (2016)
30. Su, Y., Sun, H., Sadler, B., Srivatsa, M., Gur, I., Yan, Z., Yan, X.: On generating characteristic-rich question sets for QA evaluation. In: EMNLP. pp. 562–572 (2016)
31. Talmor, A., Berant, J.: The Web as a knowledge base for answering complex questions. In: NAACL. pp. 641–651 (2018)
32. Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J.: LC-QuAD: A corpus for complex question answering over knowledge graphs. In: ISWC. pp. 210–218 (2017)
33. Usbeck, R., Gusmita, R.H., Axel-Cyrille Ngonga Ngomo, Saleem, M.: 9th challenge on question answering over linked data (QALD-9). In: SemDeep-4, NLIWoD4, and QALD-9 Joint Proceedings. pp. 58–64 (2018)
34. Usbeck, R., Röder, M., Hoffmann, M., Conrads, F., Huthmann, J., Ngonga-Ngomo, A.C., Demmler, C., Unger, C.: Benchmarking question answering systems. Semantic Web **10**(2), 293–304 (2019)

35. Völske, M., et al.: What users ask a search engine: analyzing one billion Russian question queries. In: CIKM. pp. 1571–1580 (2015)
36. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
37. Wu, Z., Kao, B., Wu, T.H., Yin, P., Liu, Q.: PERQ: Predicting, explaining, and rectifying failed questions in KB-QA systems. In: WSDM. pp. 663–671 (2020)
38. Yih, W.t., Richardson, M., Meek, C., Chang, M.W., Suh, J.: The value of semantic parse labeling for knowledge base question answering. In: ACL. pp. 201–206 (2016)
39. Zhang, X., Yang, A., Li, S., Wang, Y.: Machine reading comprehension: a literature review. arXiv preprint arXiv:1907.01686 (2019)