QALD-10 — The 10th Challenge on Question Answering over Linked Data

Shifting from DBpedia to Wikidata as a KG for KGQA

Ricardo Usbeck [a,d,*], Xi Yan* [a,d], Aleksandr Perevalov* [b,e], Longquan Jiang* [a,d], Julius Schulz* [a,d], Angelie Kraft [a], Cedric Möller [a], Junbo Huang [a,d], Jan Reineke [a,d], Axel-Cyrille Ngonga Ngomo [c], Muhammad Saleem [c] and Andreas Both [b,f]

[a] *Semantic Systems Group Universität Hamburg, Germany*
E-mails: ricardo.usbeck@uni-hamburg.de, xi.yan@uni-hamburg.de, longquan.jiang@uni-hamburg.de, julius.schulz@studium.uni-hamburg.de, angelie.kraft@uni-hamburg.de, cedric.moeller@uni-hamburg.de, junbo.huang@uni-hamburg.de, jan.reineke@uni-hamburg.de
[b] *Leipzig University of Applied Sciences, Germany*
E-mail: andreas.both@htwk-leipzig.de
[c] *DICE Group Universität Paderborn, Germany*
E-mails: axel.ngonga@upb.de, saleem@informatik.uni-leipzig.de
[d] *Hamburger Informatik Technologie-Center HITEC e.V., Germany*
[e] *Anhalt University of Applied Sciences, Germany*
E-mail: aleksandr.perevalov@hs-anhalt.de
[f] *DATEV eG, Germany*

Abstract. Knowledge Graph Question Answering (KGQA) has gained attention from both industry and academia over the past decade. Researchers proposed a substantial amount of benchmarking datasets with different properties, pushing the development in this field forward. Many of these benchmarks depend on Freebase, DBpedia, or Wikidata. However, KGQA benchmarks that depend on Freebase and DBpedia are gradually less studied and used, because Freebase is defunct and DBpedia lacks the structural validity of Wikidata. Therefore, research is gravitating toward Wikidata-based benchmarks. That is, new KGQA benchmarks are created on the basis of Wikidata and existing ones are migrated. We present a new, multilingual, complex KGQA benchmarking dataset as the 10th part of the Question Answering over Linked Data (QALD) benchmark series. This corpus formerly depended on DBpedia. Since QALD serves as a base for many machine-generated benchmarks, we increased the size and adjusted the benchmark to Wikidata and its ranking mechanism of properties. These measures foster novel KGQA developments by more demanding benchmarks. Creating a benchmark from scratch or migrating it from DBpedia to Wikidata is non-trivial due to the complexity of the Wikidata knowledge graph, mapping issues between different languages, and the ranking mechanism of properties using qualifiers. We present our creation strategy and the challenges we faced that will assist other researchers in their future work. Our case study, in the form of a conference challenge, is accompanied by an in-depth analysis of the created benchmark.

Keywords: Knowledge Graph Question Answering, Benchmark, Challenge, Query Analysis

[*] These authors equally contributed. Corresponding author: E-mail: ricardo.usbeck@uni-hamburg.de

1. Introduction

Research on Knowledge Graph Question Answering (KGQA) aims to facilitate an interaction paradigm that allows users to access vast amounts of knowledge stored in a graph model using natural language questions. KGQA systems are either designed as complex pipelines of multiple downstream task components [1, 2] or end-to-end solutions (primarily based on deep neural networks) [3] hidden behind an intuitive and easy-to-use interface. Developing high-performance KGQA systems has become more challenging as the data available on the Semantic Web, respectively in the Linked Open Data cloud, has proliferated and diversified. Newer systems must handle more volume and variety of knowledge. Finally, improved multilingual capabilities are urgently needed to increase the accessibility of KGQA systems to users around the world [4]. In the face of these requirements, we introduce QALD-10 as the newest successor of the *Question Answering over Linked Data* (QALD) benchmark series to facilitate the standardized evaluation of KGQA approaches.

1.1. The Rise of Wikidata in KGQA

Among the general-domain KGs like Freebase [5], DBpedia [6], and Wikidata [7], the latter has become a focus of interest in the community. While Freebase was discontinued, DBpedia is still active and updated on a monthly basis. It contains information that is automatically extracted from Wikipedia infoboxes, causing an overlap. However, Wikidata is community-driven and continuously updated through user input. Moreover, the qualifier model[1] of Wikidata allows more specific annotations of relations, which in turn allow for more complex questions (i.e., more than a single triple pattern). Hence, we expect that new KGQA benchmarks will utilize Wikidata and existing benchmarks will migrate away from Freebase and DBpedia to Wikidata. This becomes evident in the distribution of benchmarks represented in the curated KGQA leaderboard [8] and is supported by a few publications in which KGQA benchmarks have already been moved to Wikidata [9–11]. One approach is to map the Freebase topics to Wikidata items using automatically generated mappings, for subjects as well as objects of triples [9, 10]. For properties, handmade mappings are used. It is worth mentioning, that due to the structural differences between Freebase and Wikidata, some of the gold standard SPARQL queries cannot be transformed, and respective questions become unanswerable. The original SPARQL queries from the CFQ [12] benchmark over Freebase were mapped to Wikidata using a multi-step approach including property mapping and entity substitution.

Table 1: Infobox for QALD-10.

Name	QALD-10
URL	https://github.com/KGQA/QALD-10
Version date and number	1.0/May 29th, 2022
Licensing and availability	MIT license, open available
Topic coverage	General domain
Source for the data	Real life questions
Purpose and method of creation and maintenance	Academic purpose, manual creation and maintenance
Reported usage	Academic purpose
Metrics	Precision, Recall, Macro F1 QALD
Use of established vocabularies	RDF, purl, geosparql, wikibase, Wikidata, XMLSchema
Language expressivity	English, German, Chinese, and Russian
Growth	Static

1.2. Multilinguality in KGQA

Several works have contributed to the extension of the multilingual coverage of KGQA benchmarks. The authors of [11] were the first who manually translated LC-QuAD 2.0 [13] (an English KGQA benchmarking dataset on DBpedia) to Chinese. However, languages besides English and Chinese are not covered and the work does not

[1] https://www.wikidata.org/wiki/Help:Qualifiers

provide a deeper analysis of the issues with the SPARQL query generation process faced when working with Wikidata. The RuBQ benchmark series [14, 15] which was initially based on questions from Russian quizzes (totaling 2,910 questions) has also been translated to English via machine translation. The SPARQL queries over Wikidata were generated automatically and manually validated by the authors. The CWQ [16] benchmark provides questions in Hebrew, Kannada, Chinese, and English, with the non-English questions translated by machine translation with manual adjustments. The QALD-9-plus benchmark [17] introduced improvements and an extension of the multilingual translations in its previous version—QALD-9 [18]—by involving crowd-workers with native-level language skills for high-quality translations from English to their native languages as well as validation. In addition, the authors manually transformed gold standard queries from DBpedia to Wikidata.

1.3. Introducing QALD-10

In this paper, we present the latest version of the QALD benchmark series—QALD-10—a novel Wikidata-based benchmarking dataset. It was piloted as a test set for the 10th QALD challenge within the 7th Workshop on Natural Language Interfaces for the Web of Data (NLIWoD) [19]. This challenge uses QALD-9-plus as training set and the QALD-10 benchmark dataset as test set. QALD-10 is publicly available in our GitHub repository.[2] The infobox in Table 1 contains more details on the benchmark. We also provide a Wikidata dump and a long-term maintained SPARQL endpoint[3] for this benchmark to foster replicable and reproducible research. In summary, we make the following contributions:

- A new complex, multilingual KGQA benchmark over Wikidata—QALD-10—and a detailed description of its creation process;
- An overview of the KGQA systems evaluated on QALD-10 and analysis of the corresponding results;
- A concise benchmark analysis in terms of query complexity;
- An overview and challenge analysis for the query creation process on Wikidata.

2. QALD-10 Challenge Description and Benchmark Introduction

The QALD-10 benchmarking dataset is a part of the QALD challenge series, which has a long history of publishing KGQA benchmarks. The benchmark was released as part of the 10th QALD challenge within the 7th Workshop on Natural Language Interfaces for the Web of Data at European Semantic Web Conference (ESWC) 2022 [19].[4] While looking at past benchmarks [17, 18, 20–27], we identified several challenges. First, the **poor translation quality** for languages other than English; see QALD-9-plus paper [17] for more details. Second, the **low complexity of the gold standard SPARQL queries**. Finally, the **weak replicability** of the KGQA experiments caused by divergence between the SPARQL query results and constantly updating versions of the used knowledge graphs (e.g., DBpedia and Wikidata). With QALD-10 benchmark dataset, we remedy all the aforementioned flaws.

All the data for the challenge can be found in our project repository.[5] The QALD-10 uses the well-established QALD-JSON format[6] [28], which is adopted by other KGQA benchmarks [17, 18, 26, 27, 29]. The overall multilingual KGQA challenge contained 806 human-curated questions, including for 412 training and for 394 test. In the following, we explain the creation process in detail.

The *QALD-10 challenge training set* includes 412 questions and the corresponding queries, which are runnable against our stable SPARQL endpoint. The SPARQL query transformation from DBpedia to Wikidata was done manually by a group of computer scientists who were the authors of the QALD-9-plus paper. As some of the queries were not transformable, the total number of questions decreased from 558 to 507 between QALD-9 and

[2] Benchmark repository: https://github.com/KGQA/QALD-10. Note, that the QALD-10 is relatively stable in terms of opened community issues over time, compare to https://github.com/ag-sc/QALD/issues which hosts previous versions of this challenge.

[3] https://skynet.coypu.org/wikidata/

[4] https://www.nliwod.org/challenge

[5] https://github.com/KGQA/QALD-10/tree/main/data

[6] https://github.com/AKSW/gerbil/wiki/Question-Answering

QALD-9-plus. The number of KGQA pairs further decreased to 412 with the introduction of our stable Wikidata endpoint. These 412 question-query pairs form the QALD-10 challenge train set.

2.1. Collection of English Natural Language Questions for the QALD-10 challenge test set

The *QALD-10 challenge test set*, in contrast, was created from scratch. In the first step, we collected 500 natural language questions in English from speakers with at least a C1-level language proficiency in accordance with the Common European Framework of Reference for Languages (CEFR) [30]. We collected equal amounts of questions from each participant to ensure that the questions are unbiased and express real-world information needs. Questions vary with respect to their complexity type, including questions with counts (e.g., *How many children does Eddie Murphy have?*), superlatives (e.g., *Which museum in New York has the most visitors?*), comparatives (e.g., *Is Lake Baikal bigger than the Great Bear Lake?*), and temporal aggregators (e.g., *How many companies were founded in the same year as Google?*).

2.2. Multilingual Translations

To tackle the *first challenge*, the translations from English to Chinese, German, and Russian were created by crowd-workers in two steps: (1) each English question was translated into the target language by two native speakers, (2) the translations from the previous step were validated by another native speaker. To reduce ambiguity, the named entities in the questions were manually annotated with their Wikidata URIs before translation. The crowd-workers were asked to follow the Wikidata label of a particular entity in their native language during the translation process. Note that 12 of the Wikidata items did not have Chinese versions so the respective questions could not be labeled accordingly. For instance, the entity "The Vanishing Half" (Wikidata ID: `wd:Q98476957`) in the question *In which year was the author of "The Vanishing Half" born?* did not have a Chinese entry.

2.3. From Natural Language Question to SPARQL Query

To tackle the **second challenge**, the final set of questions was manually transformed into complex SPARQL queries over Wikidata by a group of computer scientists with complementary skills and knowledge. The incompleteness of Wikidata regarding some of the multilingual labels and the lack of an ontology caused challenges in the SPARQL query generation process. These are listed and discussed in detail in Section 5. The gold standard answers to the questions were retrieved by querying over our own Wikidata endpoint which is also available online.

2.4. Stable SPARQL Endpoint

Due to the constant updates of KGs like Wikidata, outdated SPARQL queries or changed answers can commonly cause problems for KGQA benchmarks. KG updates usually concern (1) structural changes, e.g., renaming of properties, or (2) alignment with changes in the real world, e.g., when a state has appointed a new president. According to our preliminary analysis on the LC-QuAD 2 benchmark [13], a large number of queries are no longer answerable on the current version of Wikidata. The original dump used to create the benchmark is no longer available online.[7] As a result, the authors [31] set up an endpoint with a Wikidata dump dated 13 October 2021[8] and filtered the test set of 6046 questions down to 4211 questions for which the gold query produced a valid response. We decided to follow a similar methodology by setting up a stable Wikidata endpoint that was used to execute the SPARQL queries. Hence, we provide a long-term stable endpoint to ensure reproducibility to tackle the **third challenge**. This endpoint was also provided to the participants of the QALD-10 challenge and is archived via Zenodo [32].[9]

Note, since the QALD-10 challenge training set was created before we set up this endpoint, some answers and queries had to be changed from the original release of QALD-9-plus. That is, the QALD-9-plus original data and

[7] https://databus.dbpedia.org/dbpedia/wikidata/debug/2020.07.01
[8] https://dumps.wikimedia.org/wikidatawiki/entities/
[9] https://zenodo.org/record/7496690#.Y7QfI-zMK3I

the one used as QALD-10 challenge training data are different. Different versions are recorded as releases in the GitHub repository.[10]

3. QALD-10 Challenge Evaluation

To promote the FAIR principles (Findable, Accessible, Interoperable, and Reusable) [33] with respect to our experimental results, we utilize GERBIL QA [28][11], an open-source and publicly available online evaluation tool. We adopt the established metrics for KGQA evaluation, more specifically Precision, Recall, and the Macro F1 QALD measure.

3.1. Evaluation Metric

The F-measure is one of the most commonly used metrics to evaluate KGQA systems, according to an up-to-date leaderboard [8]. It is calculated based on Precision and Recall and, thus, indicates a system's capacity to retrieve the right answer in terms of quality and quantity [34]. However, the KGQA evaluation has certain special cases due to empty answers (see also GERBIL QA [28]). Therefore, a modification was made to the standard F-measure to better indicate a system's performance on KGQA benchmarks. More specifically, when the golden answer is empty, Precision, Recall, and F-measure of this question pair receive a value of 1 only if an empty answer is returned by the system. Otherwise, it is counted as a mismatch and the metrics are set to 0. Reversely, if the system gives an empty answer to a question for which the golden answer is not empty, this will also be counted as a mismatch. Our analyses consider both micro- and macro-averaging strategies. These two averaging strategies are automatically inferred by the GERBIL system.

During the challenge, *only the Macro F1 QALD measure was used to rank the systems*. This resulted from community requests[12] and to achieve compatibility with older QALD challenges. This metric uses the previously mentioned additional semantic information with the following exception: If the golden answer set is not empty but the QA system responds with an empty answer set, it is assumed that the system determined that it cannot answer the question. Here we set the Precision to 1 and the Recall and F-measure to 0.

3.2. GERBIL QA Benchmarking Platform

The GERBIL system was originally created as a benchmarking system for named entity recognition and linking; it also follows the FAIR principles. It has been widely used for evaluation and shared tasks for its fast processing speed and availability. Later, it was extended to support the evaluation of the KGQA systems. While adopting the GERBIL framework, the evaluation can simply be done by uploading the answers produced by a system via web interface or RESTful API.[13] Each experiment has a citable, time-stable, and accessible URI that is both human- and machine-readable. The uploaded file should follow the QALD-JSON format. The GERBIL system was set up with the QALD-10 benchmark and provides an easy-to-use configuration, which allows one to choose a language for the multilingual evaluation.[14] After choosing the language and uploading the file containing a system's predictions, the evaluation is done automatically.

To promote the reproducibility of the KGQA systems and open information access, we uploaded the test result of all systems to a curated leaderboard [8].[15] The leaderboard includes the system descriptions, standardized evaluation scores, references, and other details. Therefore, it presents state-of-the-art scores for comparison purposes.

[10] https://github.com/KGQA/QALD_9_plus

[11] https://gerbil-qa.aksw.org/gerbil

[12] https://github.com/dice-group/gerbil/issues/211

[13] https://pypi.org/project/gerbil-api-wrapper/

[14] See https://gerbil-qa.aksw.org/gerbil/config-qald for the QALD configurations on GERBIL.

[15] https://kgqa.github.io/leaderboard/wikidata/qald.html#qald-10

3.3. Participating Systems

After six registrations, five teams were able to join the final evaluation. Allowing file-based submissions rather than requiring web service-based submissions led to a higher number of submissions and fewer complaints by the participants compared to previous years. Thus, unfortunately, the goal of FAIR and replicable experiments is still unreached for KGQA. Among the participating systems, three systems papers were accepted to the workshop hosting the challenge.

QAnswer [35] is a rule-based system using a combinatorial approach to generate SPARQL queries from natural language questions, leveraging the semantics encoded in the underlying knowledge graph. It can answer questions on both DBpedia and Wikidata supporting English, French, German, Italian, Russian, Spanish, Portuguese, Arabic, and Chinese. This system, which does not require training, is run as a baseline system for our challenge due to its capacity to tackle multilingual data.

SPARQL-QA [36] is a QA system that exploits Neural Machine Translation (NMT) and Named Entity Recognition (NER) modules to create SPARQL queries from natural language questions. The MNT module translates the question into a SPARQL query template in which the KB resources are replaced by placeholders, while the NER module identifies and classifies the entities present in the question. The outputs of two modules are merged to produce a new equivalent of the original SPARQL query to be executed over Wikidata, by replacing the placeholders in the template with the corresponding named entities. An uniform input format, namely QQT, is introduced to ensure training two modules together and reduce the impact of out-of-vocabulary (OOV) words.

Shivashankar et al. [37] presented a graph-to-graph transformation-based QA system using an Abstract Meaning Representation (AMR) graph to generate SPARQL queries, leveraging its ability to represent the semantics of a natural language. For a given question, its AMR graph is generated using a pre-trained multilingual AMR parser and simplified by removing unnecessary nodes and information. All possible executable SPARQL graphs are extracted from its simplified AMR graph. The system supports English and German questions.

Baramiia et al. [38] developed a QA system that first learns to predict representations of entities and properties which are close to correct queries and far from the others. It then finds the top-*k* nearest to the correct query via Scalable Nearest Neighbors method with the dot product similarity measure. It natively supports English but can be extended to the multilingual case using Transformers trained in other languages.

Suraj Singh and Dmitrii Gavrilev[16] presented a multilingual KGQA model that first translates the questions from low-resourced languages into English using a pre-trained T5 [39] model and then searches for the answer using the DeepPavlov-based [40] ensemble. The pipeline consists of query template type classification, entity detection, entity linking, relation ranking, and query generation. It can support answering questions written in English, German, Chinese, and Russian. More detailed information regarding all of the systems is provided in the proceedings [19].

3.4. Results

All systems were evaluated on the test set of QALD-10 challenge before the challenge. Participants had to upload a file their answering system generate, upload it to the GERBIL system. which would output an URL based on submission. Participants submit the GERBIL URL with their final results. Table 2 shows the systems and their performances with links to GERBIL QA. In 2022, *SPARQL-QA* [36] won the QALD-10 challenge.

4. QALD-10 Test Set Analysis

KGQA benchmarks should be complex enough to properly stress the underlying KGQA systems and hence not biased towards a specific system. Previous studies [41] have shown that various SPARQL features of the golden SPARQL queries, i.e, the corresponding SPARQL queries to QALD natural language questions, significantly affect the performance of the KGQA systems. These features include the number of triple patterns, the number of joins

[16] Their paper is not published in the proceedings.

Table 2

Evaluation results of the challenge participants' systems.

Author (System)	Language	Macro F1 QALD	GERBIL QA Link
Borroto et al. (SPARQL-QA) [36]	EN	0.595	https://gerbil-qa.aksw.org/gerbil/experiment?id=202205200035
Baseline (QAnswer) [35]	EN	0.578	https://gerbil-qa.aksw.org/gerbil/experiment?id=202205120000
Steinmetz et al. [37]	EN	0.491	https://gerbil-qa.aksw.org/gerbil/experiment?id=202205260012
Baramiia et al. [38]	EN	0.428	https://gerbil-qa.aksw.org/gerbil/experiment?id=202205210032
Singh & Gavrilev (no publication)	EN	0.195	https://gerbil-qa.aksw.org/gerbil/experiment?id=202205210017

between triple patterns, the join vertex degree, and various SPARQL modifiers such as Limit, ORDER BY, GROUP BY etc.

In this section, we compare the complexity of the QALD-10 benchmark with QALD-9-plus with respect to the aforementioned SPARQL features. The statistics of the number of questions in different QALD series datasets is shown in Table 3. We use the Linked SPARQL Queries (LSQ) [42] framework to create the LSQ RDF datasets of both of the selected benchmarks for comparison. The LSQ framework converts the given SPARQL queries into RDF and attaches query features. The resulting RDF datasets can be used for the complexity analysis of SPARQL queries [43]. The resulting Python notebooks and the LSQ datasets can be found in our GitHub repository.[17] The complexity analysis of the selected benchmarks is presented in the next subsections.

Table 3

Statistics of the number of questions in different QALD series datasets

Q10-WD Test	Q9-Plus-DB Train	Q9-Plus-DB Test	Q9-Plus-WD Train	Q9-Plus-WD Test
394	408	150	371	136

4.1. Frequency of Modifiers

When answering complex questions, KGQA systems are required to generate corresponding complex formal (SPARQL) queries, e.g., with multiple hops or constraint filters, that can represent and allow to answer them correctly. One way to represent the complexity of queries can be represented by the frequencies of modifiers (e.g., `LIMIT` or `COUNT`) that occurred in the queries. Table 4 shows the frequencies of each modifier represented in the QALD-10 test set and the different subsets of the QALD-9-plus benchmark, respectively. For QALD-9-plus, the Wikidata-based datasets use less modifiers than their DBpedia counterparts for the same set of questions. However, the results clearly suggest that the proposed benchmark is way more complex than QALD-9-plus in terms of various important modifiers such as COUNT, FILTER, ASK, GROUP BY, OFFSET, and YEAR.

4.2. Query Feature Distribution

To measure structural query complexity, we calculate the Mean and Standard Deviation (SD) values for the distributions of three query features respectively: number of triple patterns, number of join operators, as well as joint vertex degree (see the definitions in [44]). These features are often considered as a measure of structural query complexity when designing new SPARQL benchmarks [41, 43, 44]. The corresponding results are presented in Table 5. Again, the Wikidata-based datasets of QALD-9-plus have a lower distribution mean than their DBpedia counterparts and, thus, a lower complexity in general. Compared to the QALD-9-plus test sets, the QALD-10 test has a higher variation for the number of triple patterns and the number of join operators, as well as the second largest SD for joint vertex degree. This can be interpreted as getting the correct answers for the QALD-10 benchmark might be more difficult due to a wider range of possible SPARQL queries as compared to QALD-9-Plus

[17] https://github.com/KGQA/QALD-10/tree/main/notebooks

Table 4

Frequencies of each modifier in different QALD series. Note that frequencies of modifiers with the "*" character are computed using keyword matching from SPARQL queries, while the others use the LSQ framework.

Modifier	**Q10-WD**	**Q9-Plus-DB**	**Q9-Plus-DB**	**Q9-Plus-WD**	**Q9-Plus-WD**
	Test	**Train**	**Test**	**Train**	**Test**
COUNT*	**126**	57	33	32	18
LIMIT	17	39	11	**43**	12
ORDER BY	17	36	11	**43**	12
FILTER	**74**	31	17	31	13
ASK	**60**	37	4	36	3
UNION	5	**29**	17	10	6
OFFSET	**3**	1	0	2	0
GROUP BY	**95**	19	11	12	12
HAVING*	1	**3**	2	1	2
YEAR*	**43**	6	10	20	4
NOW*	1	**3**	2	1	1

Table 5

Structural complexity measured via the distribution of the number of triple patterns, the number of joins, and vertex degrees.

Query Feature		**Q10-WD**	**Q9-Plus-DB**	**Q9-Plus-DB**	**Q9-Plus-WD**	**Q9-Plus-WD**
		Test	**Train**	**Test**	**Train**	**Test**
Number of Triple Patterns	Mean	1.605	1.728	**1.993**	1.685	1.640
	SD	1.199	0.944	1.167	**1.215**	0.998
Number of Join Operators	Mean	0.622	0.509	**0.711**	0.577	0.507
	SD	**1.123**	0.662	0.869	0.929	0.686
Joint Vertex Degree	Mean	0.889	0.941	**1.089**	0.953	0.929
	SD	1.133	1.113	1.116	1.148	**1.176**

4.3. *Query Diversity Score*

From the previous results, it is still difficult to establish the final complexity of the complete benchmark. To this end, we calculate the diversity score DS of the complete benchmark B, formally defined as follows [44].

$$DS = \frac{1}{k}\sum_{i=1}^{k}\frac{\sigma_i(B)}{\mu_i(B)} \tag{1}$$

where, μ and σ are the Mean and Standard Deviation of a given distribution with respect to the i-th feature, respectively. The k is the total number of query features analyzed in B. In this work, the query features chosen are the number of triple patterns, the number of joins, and the joint vertex degree. Table 6 shows the diversity scores (the higher the score the more complex the queries) of different QALD benchmarks. We observe that the QALD-10 test set has the highest diversity score ($DS = 1.275$) compared to the other benchmarks, hence it is a good starting point for template-based KGQA benchmark generation approaches aiming at diverse, complex, large-scale characteristics.

5. Challenging Translation of Natural Language Question to Wikidata SPARQL queries

During the creation of QALD-10 test SPARQL queries for given natural language questions, we identified several challenges. Below, we systematically classify the problems into seven categories: (1) the ambiguity of the questions'

Table 6
Query diversity score of different QALD benchmarks.

Q10-WD Test	Q9-Plus-DB Train	Q9-Plus-DB Test	Q9-Plus-WD Train	Q9-Plus-WD Test
1.275	1.010	0.944	1.178	1.075

intention, (2) incompleteness of Wikidata, (3) ambiguity of SPARQL queries, (4) limit on returned answers, (5) special vocabulary, (6) calculation limitation of SPARQL, and (7) endpoint version change. We discuss the cause of these issues and present our solutions.

5.1. Ambiguity of the Natural Language Question

The question *What is the biggest city in the world?* could be asking for the most populous city or the geographically largest city. This is an example for an ambiguous natural language question. We tried to circumvent this type of questions by specifically, asking crowd-workers to phrase their questions precisely. After data collection, we chose the most reasonable interpretation based on real world experience, where necessary. Thus, in the example above, the question was changed to *What is the most populous city in the world?*. Some questions, however, remain vague and consequently correspond to multiple SPARQL queries. For instance, the question: *How many spouses do head of states have on average?* translates to:

```
SELECT (AVG(?spouseCount) AS ?result) WHERE {{
    SELECT (COUNT(DISTINCT ?spouse) AS ?spouseCount) WHERE {
        ?country wdt:P31 wd:Q6256 .
        ?country p:P35/ps:P35 ?hos .
        OPTIONAL {?hos wdt:P26 ?spouse }}
    GROUP BY ?hos
}}
```

where we used a *head of state*-property. However, using a *head of state*-class would also be feasible, leading to the SPARQL query below. There is no good way to make the question clearer except if one specifies the actual Wikidata elements, which is not realistic for real-world questions.

```
SELECT (AVG(?spouseCount) AS ?result) WHERE {{
    SELECT (COUNT(DISTINCT ?spouse) AS ?spouseCount) WHERE {
        ?hos wdt:P31 wd:Q48352 .
        OPTIONAL { ?hos wdt:P26 ?spouse }}
    GROUP BY ?hos
}}
```

5.2. Incompleteness of Wikidata

Due to the incompleteness of Wikidata, some of the entity labels do not translate to all languages of interest (see Section 2.2). To avoid unanswerable questions, we supplemented the online Wikidata by manually adding a translation approved by a linguist. However, this update is not shown in our stable endpoint which has a fixed Wikidata dump. Consequently, the labels are still missing but the questions-query-answer tuples are inserted into QALD-10 benchmark dataset.

Another issue that can arise from Wikidata's incompleteness is that data necessary to answer specific SPARQL queries can be missing. In this case, the answer can not be retrieved by a correct SPARQL query since there are no suitable triples available directly. In our benchmark, questions of this category were deleted.

5.3. Ambiguity of SPARQL Queries in Wikidata due to Ranking of Properties

Ambiguities of SPARQL queries in Wikidata are connected to the ranking mechanism of Wikidata. In Wikidata, a ranking mechanism exists which allows to annotate statements with preference information.[18] Such ranks decide

[18] https://www.wikidata.org/wiki/Help:Ranking

how relevant the values of a statement are, which becomes a source of inaccuracy when there are multiple intuitively correct ways of writing a query. In some cases, only results with high ranking will be kept while the result discards the low-ranking but correct ones. Here, we show the differences between using "`p/ps`"[19] and "`wdt`"[20] in the queries:

- The "`wdt`" prefix for properties only returns values of properties with preferred rank, if one exists. If no ranking exists, it returns every property.
- The "`p/ps`" combination always returns all properties and their values, without respecting ranks.

This makes "`wdt`" the go-to choice to find the most recent value of well-maintained properties, like head of state, which (most of the time) has the preferred rank reserved for the active head of state. The ranking mechanism also introduces semantic errors if the ranks get modified after creating the query. The ranking mechanism as a speciality of Wikidata challenges KGQA systems even more.

When a question gets complex, it is problematic to guarantee the correctness and completeness of its query e.g., *Which businesses are founded by the person in charge of Tesla?*. Here, our intuitive solution would be:

```
SELECT DISTINCT ?result
WHERE {
    wd:Q478214 wdt:P169 ?founder . # CEO of Tesla
    ?result wdt:P112 ?founder ; # founded by
        wdt:P31/wdt:P279* wd:Q4830453 . # of type business or its subclasses
}
```

Despite the exact term "business" being signified, in QALD-10 benchmark dataset, we use a property path in the last BGP [21] that generalizes its denotation to include more general instances. For example, the entity `wd:Q28222602` is also a company founded by the Tesla founder but is not linked to the Wikidata's entity business (`wd:Q4830453`). Another solution would be using `UNIONS` to find every eligible entity, in the understanding of how this task looks for an average person. In general, creating query seems impossible due to no adherence to strict ontologies in Wikidata. However, that would increase the burden on KGQA systems which learn from the annotated SPARQL queries. This problem is more severe in writing-, music- and book-related queries.

5.4. *Limit on Returned Answers*

The limit on returned answers is a significant problem in creating this dataset. A substantial amount of questions have a limited result set due to Wikidata's factual base. For instance, the question: *List the novels that won the Modern Library 100 Best Novels (`Q671613`)?* has one answer although one would expect 100 results with common sense. As a result, we discarded such questions from QALD-10 test set to maintain a righteous benchmark.

```
SELECT DISTINCT ?ans WHERE {
  ?ans wdt:P166 wd:Q671613 .
  }
```

5.5. *Special Characters*

A number of questions are based on special characters . For instance questions, *Find all Turkish verbs ending with "uş" in their lemma.* and *When did the district of Höxter come into existence?* have special characters or ask for special Wikidata properties (see example below). We tried to keep those kinds of questions with their corresponding queries as much as possible to foster multilingual KGQA research.

```
SELECT ?result
WHERE {?result
wikibase:lexicalCategory wd:Q24905;
dct:language/wdt:P218 'tr';
wikibase:lemma ?lemma.
FILTER(regex(?lemma, "uş$"))}
```

[19] `PREFIX p: <http://www.wikidata.org/prop/>,PREFIX ps: <http://www.wikidata.org/prop/statement/>`
[20] `PREFIX wdt: <http://www.wikidata.org/prop/direct/>`
[21] https://www.w3.org/TR/sparql11-property-paths/

5.6. Computational Limitations in SPARQL

SPARQL has limited capabilities to deal with numbers. For instance, there is a *lack of native normalization* for numeric values in Wikidata. For instance, a comparison or SPARQL result modifier like below does not take units into account:

```
?result wdt:P2048 ?h (..) ORDER BY DESC(?h)
```

Therefore, 100 centimeters could be bigger than 10 meters. The following query takes units into account but requires a more complicated structure:

```
?result p:P2048/psn:P2048/wikibase:quantityAmount ?h (..)
ORDER BY DESC(?h)
```

Hence, simple comparison queries may require a high level of expertise of the Wikidata schema, which makes the KGQA task on these questions more challenging. Also, there are *rounding errors in calculations*. For instance, corresponding to question: *How many years did Steve Jobs take the role of Apple CEO?*, the SPARQL would be:

```
SELECT ?result WHERE {
    wd:Q312 p:P169 ?ps .
    ?ps ps:P169 wd:Q19837;
        pq:P580 ?st ;
        pq:P582 ?et .
    BIND((YEAR(?et)-YEAR(?st)) AS ?result)
}
```

The answer based on common sense is “14”, however, the query execution produces the following: `?st = "01-09-1997"`, `?et = "23-08-2011"`, resulting in only 13 full years. Thus, KGQA systems without common-sense reasoning capabilities fail in parts of the QALD-10 test set.

5.7. Endpoint Version Changes

Finally, version changes in the endpoint and endpoint technology, especially when switching between the graph stores HDT [45] and Fuseki [22], can result in different answer sets. This is due to syntax errors and execution timeouts in their internal optimizations such as basic graph pattern[23] reordering or `rdf:type` indexing. Providing a stable endpoint in connection with a stable dump and dataset, see Section 2, helps to alleviate this challenge.

We formulate our challenges and solutions during the SPARQL generation process to aid further research in KGQA dataset creation as well as Wikidata schema research.

6. Summary

The QALD-10 benchmarking dataset is the latest version of the QALD benchmark series that introduces a complex, multilingual and replicable KGQA benchmark over Wikidata. We increased the size and complexity over existing QALD datasets in terms of query complexity, SPARQL solution modifiers, and functions. Also, we presented the issues and possible solutions while creating SPARQL queries from natural language. We have shown how QALD-10 has solved three major challenges of KGQA datasets, namely poor translation quality for languages other than English, low complexity of the gold standard SPARQL queries, and weak replicability. We deem solving the migration to Wikidata issue an important puzzle piece to providing high-quality, multilingual KGQA datasets in the future.

We were able to prove the appropriateness and robustness of the dataset by means of an ESWC challenge. Due to a pull request from the participant group,[24] we published two releases: the original QALD-10 challenge dataset in version 1.0 and an open upstream branch. Overall, the feedback of the participants on the dataset was positive.

[22] https://jena.apache.org/
[23] https://www.w3.org/TR/sparql11-query/#BasicGraphPatterns
[24] https://github.com/KGQA/QALD-10/pull/6

In the future, we will focus on generating and using existing complex, diverse KGQA datasets to develop large-scale KGQA datasets with advanced properties such as generalizability testing [46, 47] to foster KGQA research.

Acknowledgements

This work has been partially supported by grants for the DFG project NFDI4DataScience project (DFG project no. 460234259) and by the Federal Ministry for Economics and Climate Action in the project CoyPu (project number 01MK21007G). We thank also Michael Röder for supporting the GERBIL extension https://gerbil-qa.aksw.org/gerbil/config-qald.

References

[1] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix and C. Lange, Qanary–A Methodology for Vocabulary-Driven Open Question Answering Systems, in: *European Semantic Web Conference*, Springer, 2016, pp. 625–641.

[2] D. Diefenbach, K. Singh and P. Maret, WDAqua-Core1: A Question Answering Service for RDF Knowledge Bases, in *WWW '18*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, pp. 1087–1091–. ISBN 9781450356404. doi:10.1145/3184558.3191541.

[3] M. Wei, Y. He, Q. Zhang and L. Si, Multi-Instance Learning for End-to-End Knowledge Base Question Answering, *CoRR* **abs/1903.02652** (2019). http://arxiv.org/abs/1903.02652.

[4] A. Perevalov, A.-C.N. Ngomo and A. Both, Enhancing the Accessibility of Knowledge Graph Question Answering Systems through Multilingualization, in: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, 2022, pp. 251–256. doi:10.1109/ICSC52841.2022.00048.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, in: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1247–1250–. ISBN 9781605581026. doi:10.1145/1376616.1376746.

[6] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, *Semantic Web Journal* (2014).

[7] D. Vrandečić and M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, *Commun. ACM* **57**(10) (2014), 78–85–. doi:10.1145/2629489.

[8] A. Perevalov, X. Yan, L. Kovriguina, L. Jiang, A. Both and R. Usbeck, Knowledge Graph Question Answering Leaderboard: A Community Resource to Prevent a Replication Crisis, in: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, European Language Resources Association, Marseille, France, 2022, pp. 2998–3007. https://aclanthology.org/2022.lrec-1.321.

[9] D. Diefenbach, T.P. Tanon, K.D. Singh and P. Maret, Question Answering Benchmarks for Wikidata, in: *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, N. Nikitina, D. Song, A. Fokoue and P. Haase, eds, CEUR Workshop Proceedings, Vol. 1963, CEUR-WS.org, 2017. http://ceur-ws.org/Vol-1963/paper555.pdf.

[10] T.P. Tanon, D. Vrandecic, S. Schaffert, T. Steiner and L. Pintscher, From Freebase to Wikidata: The Great Migration, in: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks and B.Y. Zhao, eds, ACM, 2016, pp. 1419–1428. doi:10.1145/2872427.2874809.

[11] J. Zou, M. Yang, L. Zhang, Y. Xu, Q. Pan, F. Jiang, R. Qin, S. Wang, Y. He, S. Huang and Z. Zhao, A Chinese Multi-type Complex Questions Answering Dataset over Wikidata, *CoRR* **abs/2111.06086** (2021). https://arxiv.org/abs/2111.06086.

[12] D. Keysers, N. Schärli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon, D. Tsarkov, X. Wang, M. van Zee and O. Bousquet, Measuring Compositional Generalization: A Comprehensive Method on Realistic Data, in: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020. https://openreview.net/forum?id=SygcCnNKwr.

[13] M. Dubey, D. Banerjee, A. Abdelkawi and J. Lehmann, LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia, in: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I.F. Cruz, A. Hogan, J. Song, M. Lefrançois and F. Gandon, eds, Lecture Notes in Computer Science, Vol. 11779, Springer, 2019, pp. 69–78. doi:10.1007/978-3-030-30796-7_5.

[14] V. Korablinov and P. Braslavski, RuBQ: A Russian Dataset for Question Answering over Wikidata, in: *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II*, J.Z. Pan, V.A.M. Tamma, C. d'Amato, K. Janowicz, B. Fu, A. Polleres, O. Seneviratne and L. Kagal, eds, Lecture Notes in Computer Science, Vol. 12507, Springer, 2020, pp. 97–110. doi:10.1007/978-3-030-62466-8_7.

[15] I. Rybin, V. Korablinov, P. Efimov and P. Braslavski, RuBQ 2.0: An Innovated Russian Question Answering Dataset, in: *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, R. Verborgh, K. Hose, H. Paulheim, P. Champin, M. Maleshkova, Ó. Corcho, P. Ristoski and M. Alam, eds, Lecture Notes in Computer Science, Vol. 12731, Springer, 2021, pp. 532–547. doi:10.1007/978-3-030-77385-4_32.

[16] R. Cui, R. Aralikatte, H. Lent and D. Hershcovich, Compositional Generalization in Multilingual Semantic Parsing over Wikidata, *Transactions of the Association for Computational Linguistics* **10** (2022), 937–955. https://aclanthology.org/2022.tacl-1.55.

[17] A. Perevalov, D. Diefenbach, R. Usbeck and A. Both, QALD-9-plus: A Multilingual Dataset for Question Answering over DBpedia and Wikidata Translated by Native Speakers, 2022, pp. 229–234. doi:10.1109/ICSC52841.2022.00045.

[18] R. Usbeck, R.H. Gusmita, A.-C.N. Ngomo and M. Saleem, 9th Challenge on Question Answering over Linked Data (QALD-9) (invited paper), in: *Semdeep/NLIWoD@ISWC*, 2018.

[19] M.B. Xi Yan and R. Usbeck (eds), Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022), 2022.

[20] V. Lopez, C. Unger, P. Cimiano and E. Motta, Evaluating Question Answering over Linked Data, *Journal of Web Semantics* **21** (2013). doi:10.1016/j.websem.2013.05.006.

[21] C. Unger, P. Cimiano, V. López, E. Motta, P. Buitelaar and R. Cyganiak (eds), Proceedings of the Workshop on Interacting with Linked Data, Heraklion, Greece, May 28, 2012, in *CEUR Workshop Proceedings*, Vol. 913, CEUR-WS.org, 2012. http://ceur-ws.org/Vol-913.

[22] P. Cimiano, V. Lopez, C. Unger, E. Cabrio, A.-C.N. Ngomo and S. Walter, Multilingual Question Answering over Linked Data (QALD-3): Lab Overview, in: *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, Springer, 2013, pp. 321–332.

[23] C. Unger, C. Forascu, V. Lopez, A.N. Ngomo, E. Cabrio, P. Cimiano and S. Walter, Question Answering over Linked Data (QALD-4), in: *CLEF*, 2014, pp. 1172–1180.

[24] C. Unger, C. Forascu, V. Lopez, A.N. Ngomo, E. Cabrio, P. Cimiano and S. Walter, Question Answering over Linked Data (QALD-5), in: *CLEF*, 2015. http://ceur-ws.org/Vol-1391/173-CR.pdf.

[25] C. Unger, A.-C.N. Ngomo and E. Cabrio, *6th Open Challenge on Question Answering over Linked Data (QALD-6)*, in: *Semantic Web Challenges: Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*, Springer International Publishing, Cham, 2016, pp. 171–177. ISBN 978-3-319-46565-4.

[26] R. Usbeck, A.-C.N. Ngomo, B. Haarmann, A. Krithara, M. Röder and G. Napolitano, 7th Open Challenge on Question Answering over Linked Data (QALD-7), in: *Semantic Web Challenges*, M. Dragoni, M. Solanki and E. Blomqvist, eds, Springer International Publishing, Cham, 2017, pp. 59–69. ISBN 978-3-319-69146-6.

[27] R. Usbeck, A.-C.N. Ngomo, F. Conrads, M. Röder and G. Napolitano, 8th Challenge on Question Answering over Linked Data (QALD-8) (invited paper), in: *Semdeep/NLIWoD@ISWC*, 2018.

[28] R. Usbeck, M. Röder, M. Hoffmann, F. Conrad, J. Huthmann, A.-C. Ngonga-Ngomo, C. Demmler and C. Unger, Benchmarking Question Answering Systems, *Semantic Web Journal* (2018). http://www.semantic-web-journal.net/system/files/swj1578.pdf.

[29] L. Siciliani, P. Basile, P. Lops and G. Semeraro, MQALD: Evaluating the impact of modifiers in question answering over knowledge graphs, *Semantic Web* **13**(2) (2022), 215–231. doi:10.3233/SW-210440.

[30] Council of Europe, *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*, Council of Europe, 2001.

[31] D. Banerjee, P.A. Nair, J.N. Kaur, R. Usbeck and C. Biemann, Modern Baselines for SPARQL Semantic Parsing, *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2022).

[32] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C. Ngonga Ngomo, M. Saleem and A. Both, QALD-10 Wikidata Dump, Zenodo, 2022. doi:10.5281/zenodo.7496690.

[33] M. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.O. Bonino da Silva Santos, P. Bourne, J. Bouwman, A. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. Evelo, R. Finkers and B. Mons, The FAIR Guiding Principles for Scientific Data Management and Stewardship, *Scientific Data* **3** (2016). doi:10.1038/sdata.2016.18.

[34] C.D. Manning, *Introduction to Information Retrieval*, Syngress Publishing,, 2008.

[35] D. Diefenbach, K.D. Singh and P. Maret, WDAqua-core0: A Question Answering Component for the Research Community, in: *Semantic Web Challenges - 4th SemWebEval Challenge at ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, M. Dragoni, M. Solanki and E. Blomqvist, eds, Communications in Computer and Information Science, Vol. 769, Springer, 2017, pp. 84–89. ISBN 978-3-319-69145-9. doi:10.1007/978-3-319-69146-6_8.

[36] M.A.B. Santana, F. Ricca, B. Cuteri and V. Barbara, SPARQL-QA enters the QALD challenge, *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022)* (2022).

[37] K. Shivashankar, K. Benmaarouf and N. Steinmetz, From Graph to Graph: AMR to SPARQL, *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022)* (2022).

[38] N. Baramiia, A. Rogulina, S. Petrakov, V. Kornilov and A. Razzhigaev, Ranking Approach to Monolingual Question Answering over Knowledge Graphs, *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022)* (2022).

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P.J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *J. Mach. Learn. Res.* **21** (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html.

[40] M.S. Burtsev, A.V. Seliverstov, R. Airapetyan, M.Y. Arkhipov, D. Baymurzina, N. Bushkov, O. Gureenkova, T. Khakhulin, Y. Kuratov, D. Kuznetsov, A. Litinsky, V. Logacheva, A. Lymar, V. Malykh, M. Petrov, V. Polulyakh, L. Pugachev, A. Sorokin, M. Vikhreva and M. Zaynutdinov, DeepPavlov: Open-Source Library for Dialogue Systems, in: *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, F. Liu and T. Solorio, eds, Association for Computational Linguistics, 2018, pp. 122–127. doi:10.18653/v1/P18-4021. https://aclanthology.org/P18-4021/.

[41] M. Saleem, S.N. Dastjerdi, R. Usbeck and A.-C.N. Ngomo, Question answering over linked data: What is difficult to answer? What affects the F scores?, in: *BLINK/NLIWoD3@ ISWC*, 2017.

[42] C. Stadler, M. Saleem, Q. Mehmood, C. Buil-Aranda, M. Dumontier, A. Hogan and A.-C. Ngonga Ngomo, LSQ 2.0: A linked dataset of SPARQL query logs, *Semantic Web* (2022), 1–23.

[43] M. Saleem, M.I. Ali, A. Hogan, Q. Mehmood and A.N. Ngomo, LSQ: The Linked SPARQL Queries Dataset, in: *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan and S. Staab, eds, Lecture Notes in Computer Science, Vol. 9367, Springer, 2015, pp. 261–269. doi:10.1007/978-3-319-25010-6_15.

[44] M. Saleem, G. Szárnyas, F. Conrads, S.A.C. Bukhari, Q. Mehmood and A.-C. Ngonga Ngomo, How Representative Is a SPARQL Benchmark? An Analysis of RDF Triplestore Benchmarks, in: *The World Wide Web Conference*, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1623–1633–. ISBN 9781450366748. doi:10.1145/3308558.3313556.

[45] J.D. Fernández, M.A. Martínez-Prieto, C. Gutierrez, A. Polleres and M. Arias, Binary RDF representation for publication and exchange (HDT), *J. Web Semant.* **19** (2013), 22–41. doi:10.1016/j.websem.2013.01.002.

[46] L. Jiang and R. Usbeck, Knowledge Graph Question Answering Datasets and Their Generalizability: Are They Enough for Future Research?, in: *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J.S. Culpepper and G. Kazai, eds, ACM, 2022, pp. 3209–3218. doi:10.1145/3477495.3531751.

[47] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan and Y. Su, Beyond IID: three levels of generalization for question answering on knowledge bases, in: *Proceedings of the Web Conference 2021*, ACM, pp. 3477–3488.