

WikiKGQA 2026: The First Wiki-Based Knowledge Graph Question Answering Challenge

David Maria Schmidt^{1,*†}, Shakeeb Arzoo^{2,†}, Debayan Banerjee³, Alberto Moya Loustaunau⁴, Ricardo Usbeck³, Philipp Cimiano¹ and Aidan Hogan⁴

¹*Semantic Computing Group, CITEC, Technical Faculty, Bielefeld University, Bielefeld, Germany*

²*Crisil Ltd., S&P Global Ratings*

³*AI & Explainability, Leuphana University Lüneburg, Germany*

⁴*DCC, Universidad de Chile, Chile*

Abstract

Knowledge Graph Question Answering (KGQA) has gained significant attention from both industry and academia over the past decade. With the increasing usage of LLMs as end-to-end QA systems, their weaknesses also became clear, most importantly hallucinations and updatability. KGQA offers a promising alternative by obtaining truthful answers from trusted and well-maintained knowledge sources, such as Wikidata, without losing the flexibility of natural language input. Nevertheless, the creation of large high-quality KGQA benchmarks remains a challenge. Therefore, building on QAWiki, our goal is to turn the creation of KGQA benchmarks into a collaborative effort, just like Wikidata itself. This allows benchmarks to evolve, offering a platform for the KGQA community to propose and discuss emerging problems. As a starting point, we have invited the KGQA community to participate in a challenge based on QAWiki. Using QAWiki as a training dataset, we have developed a private test dataset with 150 benchmark items used for evaluation, featuring questions in English and Spanish, with the optional use of hand-crafted entity and relation mentions. For this test set, the queries and results have not been released to challenge participants, but have been used by the organizers to score submissions. These instances will be integrated into QAWiki after the challenge. We received submissions from five teams, and report the results of their systems in this paper. With more iterations of this competition, we hope to build both a large, high-quality KGQA benchmark as well as a sustainable community with the mission of improving it.

Keywords

Knowledge Graph Question Answering, Question Answering over Linked Data, Text-to-SPARQL, SPARQL Query Generation

1. Introduction

Among the many research directions that have emerged from the Semantic Web discourse, KGQA has garnered a lot of attention from both academia and industry. The rapid advancement of LLMs, together with the evolution of Wikidata as a trusted knowledge base of structured facts (over a billion RDF triples), provides a compelling setting to advance the state-of-the-art. In this regard, the Question Answering over Linked Data (QALD) series [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] has long served as a benchmark resource for assessing progress among different KGQA approaches. At the same time, benchmark datasets in this space exhibit well-known limitations [11], primarily due to the manual cost of creating a parallel corpus of natural language questions and structured queries. To address these issues, QAWiki [12] proposes a

Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA), co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25-29, 2026

*Corresponding author.

†These authors contributed equally.

✉ daschmidt@techfak.uni-bielefeld.de (D. M. Schmidt); arzoo.ebbs@gmail.com (S. Arzoo); debayan.banerjee@leuphana.de (D. Banerjee); amoya@dcc.uchile.cl (A. M. Loustaunau); ricardo.usbeck@leuphana.de (R. Usbeck); cimiano@techfak.uni-bielefeld.de (P. Cimiano); ahogan@dcc.uchile.cl (A. Hogan)

🌐 <https://davidmariaschmidt.de/> (D. M. Schmidt); <https://aidanhogan.com/> (A. Hogan)

🆔 0000-0001-7728-2884 (D. M. Schmidt); 0009-0003-7369-1025 (S. Arzoo); 0000-0001-7626-8888 (D. Banerjee); 0000-0002-7003-5087 (A. M. Loustaunau); 0000-0002-0191-7211 (R. Usbeck); 0000-0002-4771-441X (P. Cimiano); 0000-0001-9482-1982 (A. Hogan)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

collaboratively-edited community resource of hand-crafted question-query pairs, currently focusing on English & Spanish questions answered through Wikidata’s SPARQL service.

Based on QAWiki¹, we have organized the Wiki-based Knowledge Graph Question Answering Challenge (WikiKGQA), co-located with ISWC 2026. The motivation for this challenge is twofold: (1) to evaluate submitted KGQA systems over a hand-crafted collection of multilingual question–query pairs over Wikidata, and (2) to build a community centered around QAWiki, to expand and improve it, and thus expand and improve the resources available for training and benchmarking of KGQA systems. In this paper, we describe the challenge and provide an overview of its results. Papers published as part of these proceedings provide further details on the individual submissions [13, 14, 15, 16, 17].

2. Challenge Description

The WikiKGQA challenge (2026 edition) was conducted on four different tracks: questions in English and Spanish, each with and without mentions. A participant could choose to engage in any or all four of the tracks. The training and development data comprised a set of questions in natural language (English and Spanish), with and without the accompanying entities and relations to the Wikidata knowledge graph annotated, along with its gold standard SPARQL queries to fetch the correct answers. The source dataset was QAWiki [18] while the Codabench platform² was used to facilitate the participation, development, and evaluation of KGQA systems.

To test the performance and accuracy of the participating systems, a private dataset (withheld until after the final submission deadline of the competition) was created, containing 150 questions in English and Spanish, with half of them being annotated with their mentions data. By allowing the use of annotated mentions in two of the tracks, it became possible to generate comparative assessments on the role of entity linking, which is widely regarded as a critical component in knowledge graph question answering and query generation. The question sets for the with- and without-mentions tracks were kept disjoint to ensure the mentions data could not be used to get better results in the dedicated without-mentions track.

The performance of systems in terms of retrieving the correct results from the Wikidata knowledge graph as per the question and the ground truth SPARQL was measured based on the QALD-F1 metric³. The expected submission protocol was an array of QIDs, PIDs, or literals expressed as strings or Boolean values that can be automatically verified by the evaluation script on Codabench. Since Wikidata is a continuously evolving graph database, a static snapshot of Wikidata from December 2025⁴ was used for evaluation and also provided to challenge participants as a QLever [19] SPARQL endpoint. All datasets, along with test and submission data, are released under a CC0 license to enable reuse and reproducibility.

3. Benchmark Creation

The challenge test set was created manually by the challenge organizers. Some test questions were based on questions from QAWiki (the training set), while others were based on, e.g., properties of randomly-chosen Wikidata entities or facts that we found hard to fetch from Wikidata. Special focus was given to composing questions that are concisely expressed in natural language but whose logical counterparts are complex, multi-hop queries resulting in an unambiguous list of entities over the Wikidata endpoint. In training and test sets, we included only those queries that return one column and at least one result. For the mentions annotations, we created a simple web tool⁵ to ease the annotation

¹<https://qawiki.org>

²<https://www.codabench.org/competitions/15358/>

³<https://www.nliwod.org/challenge#evaluation-metric>

⁴<https://doi.org/10.5281/zenodo.22661413>; the main data is timestamped to 2025-12-29T23:00:38Z and the lexemes to 2026-01-02T23:00:32Z.

⁵<https://github.com/dvs23/qawiki2qaldjson>

process as well as to avoid, e.g., simple formatting errors. In addition, a dedicated Google Apps Script web application was developed to support bilingual annotation and the validation workflow: it provides a structured interface for adding and editing mention annotations, enforces the annotation rules in real time (e.g., verifying that the mention string appears verbatim in the question), and integrates directly with the shared Google Sheets used to coordinate the organizers’ work.

3.1. Mentions Annotation

To represent the mentions annotations, we developed a JSON representation corresponding to the way QAWiki represents mentions. Each such annotation contains a `string` and a `language` key, representing the annotated string from the question as well as the corresponding language, respectively. All other keys depend on the kind of mention that was annotated. If the respective substring corresponds to a Wikidata property, that property is stored using the `property` key. Similarly, if that string additionally or only references an entity, the `entity` key is used. The way a property corresponds to its textual representation can be further specified using different keys, including *inverse* (*true/false*), *superlative* (*min/max*), *quantifier* (*exists/not exists*), and *mention* (*implicit*).⁶

Several annotation guidelines were established during the creation of the test set. First, the string field must appear exactly as a contiguous substring of the question — the annotation tool enforces this as a hard validation rule. However, to accommodate cases where the relevant words are separated by intervening terms (more common in Spanish due to flexible word order), we adopted the convention of using an asterisk (*) as a wildcard within the string (e.g., “*último* superviviente*”); this occurs in 4 annotations across 2 Spanish questions and is absent from the English annotations. Second, the organizers decided against annotating property path expressions such as `wdt:P31/wdt:P279*`; only the semantically central property was annotated, and ubiquitous structural properties such as `P279` (*subclass of*) are generally left without mentions. Third, the *inverse* flag refers to the direction of the *mention* relative to the property, not to the query: for example, the word “parent” in a question would be marked *inverse* if the underlying Wikidata property is “child”, regardless of how the query is written.

The 75 questions in the with-mentions track contain 223 mention annotations in English and Spanish (identical distribution across both languages), averaging 2.97 per question (std. 1.09, min 2, max 7). Table 1 shows the distribution by mention type.

Table 1

Distribution of mention types in the WikiKGQA test dataset (with-mentions track, $n = 223$; counts are identical for EN and ES).

Mention Type	Count
Direct property	73
Entity	72
Implicit property	57
Inverse property	9
Superlative (max)	7
Superlative (min)	2
Existence	1
Non-existence	1
Specific value	1
Total	223

Table 2

Numbers of queries using certain SPARQL features in the WikiKGQA test dataset.

Feature	Total	w/ ment.	w/o ment.
SELECT	141	72	69
DISTINCT	99	46	53
PATH EXPRESSIONS	85	42	43
FILTER	23	7	16
COUNT	13	6	7
ORDER BY	11	7	4
MINUS	10	5	5
SUBQUERY	9	7	2
LIMIT	9	6	3
YEAR	9	4	5
ASK	9	3	6
GROUP BY	6	4	2
MAX	6	5	1
BIND	2	1	1
FILTER NOT EXISTS	3	0	3
HAVING	3	2	1
MIN	1	1	0
UNION	1	0	1
NOW	1	1	0

Table 3

Structural complexity and diversity statistics for the WikiKGQA test dataset. All: $n = 150$; per-track columns show mean (or median[†]) for $n = 75$ each.

Feature	Total		w/ ment.		w/o ment.	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Triples per query	2.61	1.91	2.37	1.28	2.85	2.37
Variables per query	1.79	1.19	1.65	0.89	1.92	1.41
Question length EN (chars)	55.23	19.02	52.35	16.47	58.12	20.87
Question length ES (chars)	59.77	20.53	57.53	17.73	62.01	22.77
Answer size [†] (rows, excl. ASK)	2 (median)		2 (median)		8 (median)	
Unique predicates	165		107		96	

3.2. Query Feature Distribution

Table 2 presents the number of queries involving specific SPARQL operators. Nearly all queries use SELECT (141 of 150, with the remaining 9 being ASK queries), and two thirds employ DISTINCT (99 of 150), reflecting the predominantly entity-retrieval nature of the benchmark. More complex features appear in a minority of queries: FILTER in 23, and aggregation constructs such as COUNT, GROUP BY, and HAVING in 13, 6, and 3 queries respectively. Property path expressions⁷ are used extensively (85 unique queries) to navigate through Wikidata’s complex class and instance linkages. In total, 50 queries included at least one of those SPARQL keywords (excluding ASK). Of those, 25 contain one keyword, 16 contain two different keywords, seven queries three keywords and two queries even four keywords.

Table 3 presents structural and linguistic statistics for the test set. The queries are relatively compact, with an average of 2.61 triple patterns (std. 1.91) and 1.79 variables (std. 1.19), though the standard

⁶A detailed description of the mentions annotations in QAWiki can be found here: <https://qawiki.org/wiki/QAWiki:Guide/Mentions>. The script converting QAWiki mentions to the challenge JSON format can be found here: <https://github.com/dvs23/qawiki2qaldjson>

⁷<https://www.w3.org/TR/sparql11-property-paths/>

deviations indicate considerable variation in query structure. Spanish questions are slightly longer than their English counterparts (59.77 vs. 55.23 characters on average), consistent with the idiomatic nature of Spanish questions rather than systematic verbosity. The median answer size across non-ASK queries is 2 rows overall, and also 2 for the with-mentions track, but rises to 8 for the without-mentions track, suggesting a slightly higher proportion of entity-set queries in that track; the mean is substantially higher than the median in both cases due to a small number of open-ended queries whose answers match large portions of the knowledge graph. Finally, the 165 unique predicates used across 150 queries indicate broad thematic coverage of the Wikidata property space. All queries project exactly one variable, and none use `OPTIONAL`, reflecting the entity-centric nature of the benchmark: questions target a specific entity or set of entities rather than partially-structured records.

4. Challenge Evaluation

In this section we briefly describe the evaluation metrics used for the challenge and the participating systems. After that, we present the challenge results.

4.1. Evaluation Metric

To compare the participating systems and determine a winner in each track, we used the established QALD-*F1* metric⁸. Compared to the regular definition

$$F1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

QALD-*F1* only differs by a few edge cases:

- If both the gold answer set and the prediction answer set are empty, the score is set to 1
- If the gold answer set is empty, but the prediction answer set is not empty, the score is set to 0
- If the prediction answer set is empty, but gold answer set is not empty, the score is set to 0

Since both training and test question–query pairs were limited to those generating non-empty answer sets, only the latter point is pertinent to this challenge. We consider the Macro variant of this measure. In compliance with how the Macro QALD-*F1* is usually calculated⁹, we calculated precision and recall separately for each question. The average of these single precision and recall scores is used to obtain the respective Macro scores. This avoids questions with many results being given higher weight than other questions. The Macro QALD-*F1* is then calculated as described above based on macro precision and recall.

4.2. Participating Systems

We received submissions from five teams, presented here in alphabetical order; we refer to their respective papers for more details:

- Kenmegne Fokam & Tela Tela [13] adopted a prompting-based few-shot approach using open-source models. A given test question is vectorized using TF-IDF and is subsequently matched with the top three candidate questions from the training data. Using the annotated mentions of the two tracks, the LLMs generate multiple candidate answers at different temperature settings, and the best one is selected as to whichever returns at least one result.

⁸Definition: <https://www.nliwod.org/challenge#evaluation-metric>. See <https://github.com/debayan/wikikgqa-public> for the actual implementation used for the challenge.

⁹See <https://github.com/dice-group/gerbil/issues/211#issuecomment-334168618> and <https://github.com/dice-group/gerbil/issues/320> that describe the reasoning behind the decision against calculating the mean of single *F1* scores.

Table 4

WikiKGQA 2026 test set results; only best submission per team is shown.

Team	Best Macro QALD-F1
English with mentions	
Mihindukulasooriya et al. (ASTER)	0.88
Duan et al. (AMA-KBQA)	0.87
Kenmegne Fokam & Tela Tela (P3)	0.65
Marian et al. (WikiProSP)	0.63
English without mentions	
Walter & Bast (GRASP/GRISP)	0.85
Mihindukulasooriya et al. (ASTER)	0.81
Duan et al. (AMA-KBQA)	0.79
Spanish with mentions	
Kenmegne Fokam & Tela Tela (P3)	0.63
Spanish without mentions	
Walter & Bast (GRASP/GRISP)	0.83

- Duan et al. [14] used the AMA-KBQA framework [20] for their submission. This multi-agent KGQA framework uses a ReAct-style loop in combination with typed Model Context Protocol tools that wrap SPARQL requests. In addition to the reasoning core that remained unchanged, they added multiple components to adapt to the WikiKGQA challenge, including an answer-convention alignment layer to comply with the expected answer types of the challenge.
- Marian et al. [15] compare three LLM-based methods with two retrieval-based methods, overall ranging from zero-shot over few-shot with automated repairs to TF-IDF and BM25. The retrieval-based approaches compare the input question with question-SPARQL pairs from the training set and return the most similar example. The LLM-based approaches let an LLM generate the final query, both with and without examples from the training set as well as with and without mentions annotations. As another variant, the final query is passed back to the LLM to let it review and, if necessary, repair the generated query.
- Mihindukulasooriya et al. [16] presented an agentic skill-driven approach, decomposing the SPARQL query generation process into several steps, including graph exploration, entity and relation linking, query construction, and result validation. They emphasize the focus of their approach on agentic skills as reusable procedural knowledge.
- Walter & Bast [17] present the results of running three different approaches on the WikiKGQA dataset. GRASP [21] is based on state-of-the-art centralized LLMs that explore a knowledge graph and then iteratively refine a query that would best answer a question, while GRISP [22] is a fully localized alternative that is fine-tuned to generate SPARQL queries. Finally, the results of the two systems are compared with Claude Code’s off-the-shelf solution. All experiments conducted with the test dataset are documented covering runtime, financial costs, and the results of fine-tuning the available datasets.

4.3. Results

The five different teams participated across the four tracks of the challenge. Four of them participated in the English track with mentions, three in the English track without mentions, as well as one each in the Spanish tracks with and without mentions annotations. The best results of each team are presented in Table 4. Thus, the winners of the English track with mentions are Mihindukulasooriya et al. with a Macro QALD-F1 score of 0.88, Kenmegne Fokam & Tela Tela won the Spanish track with mentions with 0.63, and Walter & Bast won both tracks without mentions achieving Macro QALD-F1 scores of

0.85 and 0.83, respectively.

4.4. Discussion

The systems achieved impressive results, with top scores in the interval of [0.8,0.9) demonstrating that very good performance is possible for KGQA, even on diverse, massive knowledge graphs like Wikidata. However, the task is not yet solved. We discuss two examples of challenging question-query pairs that no competing system managed to resolve. Further examples are provided in Appendix A. Such questions are at the frontier of what state-of-the-art KGQA systems are capable of. Finally, we conclude the discussion with some open questions for KGQA systems that emerged from hosting this challenge.

Example 1: Which players have won the Grand Slam in Tennis?

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT DISTINCT ?sbj1
WHERE { { wd:Q102113 wdt:P527 ?obj1 .
        ?sbj1 ^wdt:P1346 ?obj2 .
        ?obj2 wdt:P361/wdt:P31/wdt:P279* ?obj1 .
        ?obj2 wdt:P585 ?sbj2 .
        ?obj2 wdt:P2094 ?obj3 .
        BIND (YEAR(?sbj2) AS ?year) }
        MINUS { wd:Q102113 wdt:P527 ?obj1 .
        ?sbj1 ^wdt:P1346 ?obj2 .
        ?obj2 wdt:P361/wdt:P31/wdt:P279* ?obj1 .
        ?obj2 wdt:P585 ?sbj2 .
        ?obj2 wdt:P2094 ?obj3 .
        BIND (YEAR(?sbj2) AS ?year)
        wd:Q102113 wdt:P527 ?obj4 .
        MINUS { wd:Q102113 wdt:P527 ?obj4 .
        ?sbj1 ^wdt:P1346 ?obj5 .
        ?obj5 wdt:P361/wdt:P31/wdt:P279* ?obj4 .
        ?obj5 wdt:P585 ?sbj3 .
        ?obj5 wdt:P2094 ?obj3 .
        BIND (YEAR(?sbj3) AS ?year) } } }
```

The question and its equivalent query are difficult to frame exactly, although a complete list of players who have accomplished this feat is available over Wikidata. None of the participating systems came close to resolving the correct form of the query and getting the right list of entities from the endpoint. The results can be verified over the Wikipedia page, Grand Slam (tennis).¹⁰

Example 2: Who was the last surviving World War I veteran?

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?sbj
WHERE { ?sbj wdt:P607 wd:Q361 .
        ?sbj wdt:P31 wd:Q5 .
        ?sbj wdt:P570 ?obj .}
ORDER BY DESC(?obj)
LIMIT 1
```

In this example, the “true” answer is not obtained through the gold standard query. The factually correct answer, Florence Green (Q236962), is erroneously linked in Wikidata, which results in its

¹⁰[https://en.wikipedia.org/wiki/Grand_Slam_\(tennis\)](https://en.wikipedia.org/wiki/Grand_Slam_(tennis))

exclusion from the gold standard entities. At the time of writing this article, subsequent changes in Wikidata lead to a situation where running the same query returned Hugh Toye (Q21151381), whom Wikidata falsely designates as a World War I veteran rather than a World War II veteran. Similarly, because the gold standard query does not exclude unknown death dates, the gold standard result, Feliks Dziewicki (Q134695300) is incorrect as well. But even if the query accounted for unknown death dates, the fixed Wikidata dump would have returned James Gowan (Q20643094), which is another mix-up of World War I and II. Despite the fact that the answer based on a “correct” SPARQL from a static Wikidata dump is incorrect, there were participating systems that retrieved the factually-correct entity, Florence Green (Q236962), most likely by referring to its entity description as the “*last surviving veteran of World War I*”. This raises an interesting question as to what is expected from KGQA systems. For WikiKGQA, we take as the expected answer whatever the fixed version of Wikidata generates for the (correct) gold standard query, irrespective of whether the answer is correct or not. Undoubtedly, from a user’s perspective, it is getting the factually “correct” answers. However, from the point of view of knowledge organization and representation, it is debatable whether unstructured textual information, i.e., entity descriptions, or some form of parametric knowledge within KGQA systems are more important than the precise operations of logically posed queries.

Other open issues for KGQA Apart from this error analysis, two other open questions – as also highlighted by Walter & Bast [17] – important for KGQA relate to:

- the computational and financial cost of achieving high *F1* results, where there is an evident trade-off between the quality of queries/results returned, the time taken (e.g., adding more steps or more inference time), and the overall financial cost (e.g., paying for a more powerful model, for more time, with higher priority, etc.); indeed, for state-of-the-art KGQA, developers and users must choose between the speed, cost, and quality of features and services.
- the fact that generic baselines based on Claude achieve competitive results with KGQA-specific approaches [17], raising questions about the role and benefit of KGQA-specific systems in future.

These aspects merit further emphasis in future challenges.

5. Future Directions for WikiKGQA

There are multiple aspects of the challenge that we would like to improve in future editions. Some of these aspects are listed in the following sections.

5.1. Expected Answer Specification

This year’s edition of WikiKGQA had a rather implicit definition of what was expected as an output for a specific question. Although the focus on entities instead of, e.g., labels and names was comparably clear, for some edge cases there could be non-obvious deviations from the gold standard query leading to low scores in those cases although the query in principle was a plausible representation of the input question. For future challenges, we thus aim to publish specifications of the expected answers together with the questions to make expectations clear for all participants and avoid penalties due to misunderstandings.

5.2. New or Unseen KG

Wikidata is very well-suited for KGQA challenges like WikiKGQA, especially due to the extensive and diverse knowledge stored there. However, due to the popularity and size of Wikidata, it is also part of the training data of most current LLMs in one way or another, thus potentially having substantial impact on the performance of LLM-based KGQA systems. Although this makes the challenge accessible for a broad audience and potential participants, it also may hide some of the differences in terms

of generalization and set-up requirements of participating systems. For future challenges, we will therefore consider additionally providing a track with a novel or less-known KG in combination with corresponding questions. However, it remains to be explored what that means for the effort necessary to host such a query and to provide the training/test sets.

5.3. Considering Efficiency and Costs

In today’s broad landscape of diverse LLMs, LLM-based approaches, as well as KGQA systems in general, results in KGQA systems that may substantially differ in terms of required computation power during training and/or inference, as well as wall-clock time from asking a question to the final generated query. In this year’s WikiKGQA edition, this aspect was not considered for the comparison of participating systems. The decision to compare systems based on results submitted by the participants instead of submitted KGQA systems that are then evaluated on the test dataset by the organizing committee was mostly made due to the constrained time and computation resources available. As we also see the necessity to take system efficiency into account to compare KGQA systems in a fair way, we will explore different possibilities to include this into WikiKGQA in future editions: clearly a KGQA system that does not “win” on *F1* may still deserve much credit if it is competitive on *F1* and much cheaper, or more efficient. Of interest is that, though not requested, some participants opted to include such analysis in their submissions – including Walter & Bast [17], Kenmegne Fokam & Tela Tela [13], as well as Duan et al. [14] – which provided valuable insights into the comparable efficiency of different approaches and models.

5.4. Improving Mentions Annotations

For this year’s with-mentions tracks, we closely followed the way QAWiki annotates mentions while developing the test sets. However, we implicitly constrained most test set annotations of mentions to a) sub-strings of the input question and b) entities and properties that occur in the gold standard query. In order to make the usage of mention information as easy as possible for participants, we plan to provide more detailed descriptions of their structure and meaning in the future. Moreover, the JSON format we used in the challenge to provide mentions annotations could also benefit from a detailed documentation with examples. Similarly, we also plan to further abstract the mention annotations to be represented with OntoLex/Lemon [23] lexical entries, which is already the format used for Wikidata lexemes. However, Wikidata lexemes are not yet connected to the corresponding properties they denote. This is where in the future we could contribute to extend Wikidata correspondingly.

5.5. Question Ambiguity

Although we aimed to avoid overtly ambiguous questions during dataset creation, this is a continuous rather than a discrete criterion, where we do not have a fixed measure or threshold. Post publication of datasets, we encountered at least two potential cases of entity ambiguity: (1) “*Who wrote The Lost World?*”, where there are two famous novels with the same name, one by Michael Crichton (Q580361), and one by Arthur Conan Doyle (Q785255); (2) “*What is the population of Denmark?*” could be interpreted as referring to Denmark (Q35), or the Kingdom of Denmark (Q756617), where only the latter includes Greenland. Thus, some questions in the training and test sets might be open to ambiguity. QAWiki [12] itself provides support for annotating ambiguous questions, though this addresses ambiguity in terms of question intent or criteria, rather than entity ambiguity (e.g., “*What is the largest country in Africa?*” is disambiguated by “*What is the largest country in Africa by area?*” and “*What is the largest country in Africa by population?*”). Disambiguation of entities is a trickier issue, since many entities potentially exist with the same name, and not all are problematic, due to the higher prior associated with a well-known entity versus lesser-known namesakes (e.g., the answer to “*In which country is Paris?*” is likely France, rather than the U.S., Canada, etc.). The boundary between problematic and unproblematic cases of ambiguity, and entity ambiguity in particular, seems somewhat arbitrary.

5.6. Quality Control

After publishing the test questions, we noticed two cases of questions with quality issues. The first question, “*Which capital city is located on the longest river in the Europe?*”, does not permit an answer. The gold-standard SPARQL query is well-expressed, and at one point it returned Vienna (Q1741) as an answer, seemingly assuming the Danube to be the river in question, whereas it is rather the Volga, which does not have a capital city situated on it. While in general, any results generated by the gold-standard query on the selected Wikidata dump are deemed “correct” for the purposes of this evaluation, enhanced quality control is required to indeed ensure correctness. In another case, for the question “*Is Ben Jonson older than William Shakespeare?*”, the gold SPARQL query was incorrect as Shakespeare is older than Jonson (1564 > 1572): the inequality sign was erroneously reversed in the gold query. Both of these cases indicate the need for us to implement better quality-control in the future, even though they had no impact on the winners of the respective tracks. Creating a KGQA gold standard by hand is labor-intensive, and prone to errors (indeed this is also a founding idea behind QAWiki [12], to make such resources better via collaborative editing and continuous improvement). Despite considerable effort to ensure a high-quality dataset, some errors still crept in. For next time, we wish to implement semi-automatic quality control, potentially using a strong KGQA system to generate answers and double-checking cases where its results differ from the gold standard.

5.7. Benchmark Difficulty

Table 4 shows that many of the competing systems achieved impressive results, and the gap between different approaches is relatively small. We believe that this may be due to there being a high number of easier questions that all systems are capable of answering, and not enough challenging questions that separate better the performance of systems. Our goal with the benchmark was to provide concise questions that are similar, in principle, to what a user might ask, and many such questions result in relatively simple queries. For a future challenge, it would be worthwhile to include a high-proportion of challenging questions to ensure a better spread of system performance.

5.8. KGQA Reverse Engineering

In our error analysis, we found cases where the models found the correct answer, even though an incorrect answer was given by the gold standard query on the static Wikidata dump. This suggests that systems may in some cases be generating queries via a form of shortcut learning rather than based (solely) on question intent. For example, a model can go through a “reverse route” and generate the query by first finding the correct solution – by looking up a textual description or retrieving it from its parametric knowledge – and reverse-engineering some query to return that solution [24]. Textual and other contextual information can be very valuable as complementary resources to build better queries. However, LLMs appear to have fundamental limitations regarding compositionality, w.r.t. SPARQL query generation [25], as well as in a more general sense [26]. From a KGQA perspective, the query representing the input question is thus a useful output *in and of itself* alongside the solutions. Evaluating queries based on solutions does not flag cases where queries return correct solutions but do not reflect question intent. This kind of behavior is not trivial to avoid or pin down in a challenge, but we will evaluate different options on how to deal with this issue in future challenges.

5.9. Dynamic Solutions

A related issue is the changing state of KGs (especially Wikidata). The error analysis indicates that there is a temporal dimension applicable to all data, where KGQA systems sometimes generated correct answers but for a different time-frame (not the fixed Wikidata version). Biographical information about entities, for example, can be contested, refined, and revised as soon as new facts emerge. To mitigate such volatility for the challenge, we used a fixed Wikidata dump. However, as discussed for a few cases, wrong or missing information within this specific snapshot can lead to partially incorrect answer sets.

This makes identifying the correct answer difficult, particularly when valid data remain present in the chosen version but are overshadowed by erroneous information. While necessary, this version-locking also introduced constraints during test dataset creation, as newly discovered errors could no longer be rectified once the endpoint was fixed. This led to some quality control issues as well as an ambiguous interpretation of what the “gold set” of entities would be. However, it remains unclear whether a superior alternative exists, as equally valid arguments support the use of a fixed Wikidata dump as the most practical way to evaluate how well each system works within the specific frame of reference.

6. Conclusions

We have presented the rationale, design, participating systems and results of the first Wiki-based Knowledge Graph Question Answering Challenge (WikiKGQA 2026), based on QAWiki. Our evaluation has demonstrated that state-of-the-art KGQA systems can achieve impressive KGQA results over large-scale, highly-diverse knowledge graphs such as Wikidata. However, the task is not solved as we also discuss examples of question-query pairs that no system could correctly solve, showing that more complex cases still lie out of reach of state-of-the-art systems. We further highlight issues relating to computational efficiency and financial costs that require further attention in future. There are many concerns regarding the maturity and robustness of current methods for benchmarking KGQA systems; this is in part due to the high effort required to ensure high-quality datasets. With the aim of repeating such a challenge in future, while learning from this experience, we have outlined nine aspects that could be improved for future occurrences of this challenge. Combined with QAWiki, we hope that this results in incremental improvements for datasets available to train and benchmark KGQA systems.

Acknowledgments

We gratefully acknowledge the contributions of those who added and improved questions on QAWiki, and thus helped to build the dataset used for training. We are also very grateful to the contestants who submitted results to this challenge and, through it, helped us to learn more about the state-of-the-art of KGQA systems.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

- [1] QALD-1 | Semantic Computing Group, 2012. URL: <https://web.archive.org/web/20120116090114/http://www.sc.cit-ec.uni-bielefeld.de/qald-1>.
- [2] C. Unger, P. Cimiano, V. López, E. Motta, P. Buitelaar, R. Cyganiak (Eds.), Proceedings of the Workshop on Interacting with Linked Data, Heraklion, Greece, May 28, 2012, volume 913 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012.
- [3] P. Cimiano, V. Lopez, C. Unger, E. Cabrio, A.-C. Ngonga Ngomo, S. Walter, Multilingual Question Answering over Linked Data (QALD-3): Lab Overview, in: D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P. Forner, H. Müller, R. Paredes, P. Rosso, B. Stein (Eds.), Information Access Evaluation. Multilinguality, Multimodality, and Visualization, volume 8138, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 321–332. doi:10.1007/978-3-642-40802-1_30.
- [4] C. Unger, C. Forascu, V. Lopez, A.-C. Ngonga, E. Cabrio, P. Cimiano, S. Walter, Question Answering over Linked Data (QALD-4) (2014).

- [5] C. Unger, C. Forascu, V. Lopez, A.-C. Ngonga, E. Cabrio, P. Cimiano, S. Walter, Question Answering over Linked Data (QALD-5) (2015).
- [6] C. Unger, A.-C. N. Ngomo, E. Cabrio, 6th Open Challenge on Question Answering over Linked Data (QALD-6), in: H. Sack, S. Dietze, A. Tordai, C. Lange (Eds.), *Semantic Web Challenges*, Springer International Publishing, Cham, 2016, pp. 171–177. doi:10.1007/978-3-319-46565-4_13.
- [7] R. Usbeck, A.-C. N. Ngomo, F. Conrads, M. Röder, G. Napolitano, 8th Challenge on Question Answering over Linked Data (QALD-8) (invited paper), in: K.-S. Choi, L. E. Anke, T. Declerck, D. Gromann, J.-D. Kim, A.-C. N. Ngomo, M. Saleem, R. Usbeck (Eds.), *Joint Proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data Challenge (QALD-9)*, volume 2241 of *CEUR Workshop Proceedings*, CEUR, 2018, pp. 51–57. URL: <https://ceur-ws.org/Vol-2241/#paper-05>.
- [8] R. Usbeck, R. H. Gusmita, A. N. Ngomo, M. Saleem, 9th challenge on question answering over linked data (QALD-9), in: *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC)*, California, United States of America, 2018, pp. 58–64.
- [9] A. Perevalov, D. Diefenbach, R. Usbeck, A. Both, QALD-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers, in: *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, 2022, pp. 229–234. doi:10.1109/ICSC52841.2022.00045.
- [10] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C. Ngonga Ngomo, M. Saleem, A. Both, QALD-10 – The 10th challenge on question answering over linked data: Shifting from DBpedia to Wikidata as a KG for KGQA (2023) 1–15. doi:10.3233/SW-233471.
- [11] L. Jiang, R. Usbeck, Knowledge graph question answering datasets and their generalizability: Are they enough for future research?, in: E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, G. Kazai (Eds.), *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, Spain, July 11 - 15, 2022, ACM, 2022, pp. 3209–3218. doi:10.1145/3477495.3531751.
- [12] A. M. Lousaunau, A. Hogan, QAWiki: A Knowledge Graph Question Answering & SPARQL Query Generation Dataset for Wikidata, in: N. Ferranti, M. Machado, J. G. Junior, S. Razniewski, A. Polleres, M. Krötzsch (Eds.), *Proceedings of the 5th Wikidata Workshop (Wikidata 2025) co-located with ISWC 2025*, Nara, Japan, November 2, 2025, volume 4108 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: <https://ceur-ws.org/Vol-4108/paper12.pdf>.
- [13] E. C. K. Fokam, G. E. T. Tela, TF-IDF Example Retrieval and Execution-Guided Selection for Few-Shot Text-to-SPARQL over Wikidata with Open-Weight LLMs, in: *Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA)*, co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25–29, 2026, CEUR, 2026. (in these proceedings).
- [14] Y. Duan, M. Klattig, M. Bardenhagen, Y. Li, T. Käfer, Answer-Convention Alignment for Agentic Text-to-SPARQL: AMA-KBQA at the WikiKGQA 2026 Challenge, in: *Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA)*, co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25–29, 2026, CEUR, 2026. (in these proceedings).
- [15] M. Marian, A. Nanu, A. Dan, G. C. Silaghi, A Prompt-Based Approach to WikiKGQA 2026: SPARQL Generation with Large Language Models, in: *Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA)*, co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25–29, 2026, CEUR, 2026. (in these proceedings).
- [16] N. Mihindukulasooriya, N. Dsouza, S. Kashyap, O. Seneviratne, H. Samulowitz, Can Skill-Driven LLM Agents Solve Text2SPARQL through Autonomous Exploration and Iterative Refinement?, in: *Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA)*,

co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25–29, 2026, CEUR, 2026. (in these proceedings).

- [17] S. Walter, H. Bast, GRASP and GRISP for Question Answering over Wikidata, in: Proceedings of the Wiki-Based Knowledge Graph Question Answering Challenge (WikiKGQA), co-located with the 25th International Semantic Web Conference (ISWC 2026), Bari, Italy, October 25–29, 2026, CEUR, 2026. (in these proceedings).
- [18] S. Arzoo, A. Hogan, A. Moya Loustaunau, D. M. Schmidt, Qawiki v2026.2: Knowledge graph question answering (kgqa) / sparql query generation dataset for wikidata, 2026. URL: <https://doi.org/10.5281/zenodo.19474009>. doi:10.5281/zenodo.19474009.
- [19] H. Bast, B. Buchhold, QLever: A Query Engine for Efficient SPARQL+Text Search, in: E. Lim, M. Winslett, M. Sanderson, A. W. Fu, J. Sun, J. S. Culpepper, E. Lo, J. C. Ho, D. Donato, R. Agrawal, Y. Zheng, C. Castillo, A. Sun, V. S. Tseng, C. Li (Eds.), Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017, ACM, 2017, pp. 647–656. URL: <https://doi.org/10.1145/3132847.3132921>. doi:10.1145/3132847.3132921.
- [20] Y. Duan, M. Klattig, Y. Li, T. Käfer, AMA-KBQA: An adaptable multi-agent framework for generalizable knowledge base question answering, in: Proc. of SEMANTiCs Posters & Demos, 2026. (to appear).
- [21] S. Walter, H. Bast, Grasp: Generic reasoning and sparql generation across knowledge graphs, in: D. Garijo, S. Kirrane, A. Salatino, C. Shimizu, M. Acosta, A. G. Nuzzolese, S. Ferrada, T. Soulard, K. Kozaki, H. Takeda, A. L. Gentile (Eds.), The Semantic Web – ISWC 2025, Springer Nature Switzerland, Cham, 2026, pp. 271–289.
- [22] S. Walter, H. Bast, GRISP: Guided Recurrent IRI Selection over SPARQL Skeletons, 2026. URL: <https://arxiv.org/abs/2604.21133>. arXiv:2604.21133.
- [23] J. P. McCrae, D. Spohr, P. Cimiano, Linking Lexical Resources and Ontologies on the Semantic Web with Lemon, in: Proceedings of the 8th extended semantic web conference on The semantic web: research and applications (ESWC), volume 6643, 2011, pp. 245–259.
- [24] M. Arenas, G. I. Diaz, E. V. Kostylev, Reverse Engineering SPARQL Queries, in: J. Bourdeau, J. Hendler, R. Nkambou, I. Horrocks, B. Y. Zhao (Eds.), Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016, ACM, 2016, pp. 239–249. URL: <https://doi.org/10.1145/2872427.2882989>. doi:10.1145/2872427.2882989.
- [25] D. M. Schmidt, R. Schubert, P. Cimiano, CompoST: A Benchmark for Analyzing the Ability of LLMs to Compositionally Interpret Questions in a QALD Setting, in: D. Garijo, S. Kirrane, A. Salatino, C. Shimizu, M. Acosta, A. G. Nuzzolese, S. Ferrada, T. Soulard, K. Kozaki, H. Takeda, A. L. Gentile (Eds.), The Semantic Web – ISWC 2025, volume 16140, Springer Nature Switzerland, Cham, 2026, pp. 3–22. doi:10.1007/978-3-032-09527-5_1.
- [26] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, B. Y. Lin, S. Welleck, P. West, C. Bhagavatula, R. Le Bras, J. Hwang, S. Sanyal, X. Ren, A. Ettinger, Z. Harchaoui, Y. Choi, Faith and fate: Limits of transformers on compositionality, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 70293–70332. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/deb3c28192f979302c157cb653c15e90-Paper-Conference.pdf. doi:10.52202/075280-3081.

A. Error Analysis

Question: <i>Which heraldic emblems feature a bear?</i>	
Gold Query: <pre>SELECT DISTINCT ?subj WHERE { ?subj wdt:P31/wdt:P279* wd:Q112110 . ?subj wdt:P180*/wdt:P279* wd:Q11788 .}</pre>	Discussion: Although the Wiki-page “bears in heraldry”¹¹ gives the impression that a longer list of entities matches the description from the question; Wikidata still contains around 16 entities featuring bears in blazonry. An analysis of the submitted queries reveal that participating systems refer to “bear” (Q1019010) as a heraldic animal, rather than at the family level (Ursidae) of the species. Artists designing a coat of arms frequently refer to the term “ursine” (bear family) to convey the bear-like features hence the entity linking aspect should ideally resolve as such.
Question: <i>What contraindications are linked with drugs used in asthma treatment?</i>	
Gold Query: <pre>SELECT DISTINCT ?obj2 WHERE { ?subj wdt:P2175 wd:Q35869 . ?subj p:P769 ?obj . ?obj ps:P769 ?obj1 . ?obj pq:P1909 ?obj2 .}</pre>	Discussion: This query hinges on the annotated mentions of “significant drug interaction” (P769) and reported “side effects” (P1909) as the target properties. The expected answers should receive a list of medical symptoms, conditions, and factors in the results. None of the submitted systems get it correct.
Question: <i>Which poets are buried in Montparnasse Cemetery?</i>	
Gold Query: <pre>SELECT DISTINCT ?subj WHERE { ?subj wdt:P106/wdt:P279* wd:Q49757 . ?subj wdt:P119 wd:Q272208 .}</pre>	Discussion: The question results in a rather straightforward multi-hop query featuring the inclusion of two constants: poet (Q49757) and Montparnasse Cemetery (Q272208). The submitted queries by the top two systems are almost correct except for excluding the property path (P279*). This would enable the inclusion of the occupation of “chansonner” (Q13391399) which is a sub-class of the “poet” occupation in the French tradition.

¹¹https://commons.wikimedia.org/wiki/Category:Bears_in_heraldry

Question: <i>Which players have achieved Career Grand Slam in Tennis?</i>	
Gold Query: <pre> SELECT DISTINCT ?subj WHERE { { wd:Q102113 wdt:P527 ?obj1 . ?subj ^wdt:P1346 ?obj2 . ?obj2 wdt:P361/wdt:P31/wdt:P279* ?obj1 . ?obj2 wdt:P2094 ?obj3 . MINUS { wd:Q102113 wdt:P527 ?obj1 . ?subj ^wdt:P1346 ?obj2 . ?obj2 wdt:P361/wdt:P31/wdt:P279* ?obj1 . ?obj2 wdt:P2094 ?obj3 . wd:Q102113 wdt:P527 ?obj4 . MINUS { ?subj ^wdt:P1346 ?obj5 . ?obj5 wdt:P361/wdt:P31/wdt:P279* ?obj4 . ?obj5 wdt:P2094 ?obj3 . } } } } </pre>	Discussion: Similar to the earlier Grand slam question but by excluding the time property (P585) and its referred variable, the top systems came close to generating the correct results. The issue with the results submitted by the GRASP [17] and ASTER systems [16] was the restriction to singles competition (both men’s and women’s tournaments) and the exclusion of doubles’ competition. The results are available in the Wikipedia page for Grand Slam.¹²
Question: <i>Which species endemic to New Zealand are critically endangered?</i>	
Gold Query: <pre> SELECT DISTINCT ?subj WHERE { ?subj wdt:P141 wd:Q219127 . ?subj wdt:P183 wd:Q664 .} </pre>	Discussion: The submitted queries get the gold set of entities right except for one missing dolphin, the Māui dolphin (Q3696045), whose taxon rank is asserted in Wikidata as a subspecies of Hector’s dolphin (Q300961). The error stems from the SPARQL variable referencing the taxon rank property (P105) to species (Q7432) [i.e., it frames the SPARQL as ?species wdt:P105 wd:Q7432 .].
Question: <i>Who was US president before Donald Trump?</i>	
Gold Query: <pre> SELECT DISTINCT ?obj2 WHERE { ?subj wdt:P2175 wd:Q35869 . ?subj p:P769 ?obj . ?obj ps:P769 ?obj1 . ?obj pq:P1909 ?obj2 .} </pre>	Discussion: A straightforward query that was erroneously handled by most systems. Only one system (WikiProSP [15]) got the correct entities, i.e., both Barack Obama (Q76) and Joe Biden (Q6279). Another system wrote the correct query but submitted only one entity as the final answer, while another hardcoded Donald Trump as the 45th president of the US, thereby getting only Q76, while the fourth submission wrote the wrong query altogether.

¹²[https://en.wikipedia.org/wiki/Grand_Slam_\(tennis\)](https://en.wikipedia.org/wiki/Grand_Slam_(tennis))