


Instruction-Tuned Language Models as Judges for SPARQL Query Correctness in Knowledge Graph Question Answering

Aleksandr Gashkov¹, Maria Eltsova^{1,2}, Aleksandr Perevalov¹ and Andreas Both^{1,**}

¹  Web & Software Engineering (WSE) Research Group, Leipzig University of Applied Sciences (HTWK Leipzig), Karl-Liebknecht-Straße 132, 04277 Leipzig, Germany

² CBZ München GmbH, Heilbronn, Germany

Abstract

Nowadays, the research community pays increasing attention to the challenge of trustworthy Knowledge Graph Question Answering (KGQA) systems due to the expectation of returning a high-quality and correct answer to the given natural-language question from continuously growing Knowledge Graphs (KGs). However, modern KGQA systems still generate a lot of incorrect SPARQL queries, leading to many incorrect answers presented to users. In this paper, we follow our long-term research agenda of providing an approach that advances the trustworthiness of KGQA systems while filtering out the incorrect query candidates (following the principle: no answer is better than a wrong answer). The approach presented in this paper is based on the use of LLMs that help to distinguish between correct and incorrect query candidates. Here, we aim to create a general approach that is, firstly, independent of the used (a) language(s), (b) KGs, (c) LLMs, and, secondly, can improve the answer quality of any KGQA system. For our experiments, we used LLMs from the following families: DeepSeek, Llama, Mistral, OpenAI, and Qwen. The LLMs were applied to the two state-of-the-art multilingual KGQA systems – QAnswer and MST5 – as post-processing SPARQL query filters. The approach was evaluated using the multilingual Wikidata-based dataset QALD-9-plus. The experimental results indicate reasonable quality improvement for all languages when using the approach presented in this paper.


Keywords

Question Answering over Knowledge Graphs, SPARQL Validation, Trustworthiness, Large Language Models, Multilingual Approach


1. Introduction

The use of large language models (LLMs) in many areas of NLP, including question answering (QA), is the recent trend in the research community (e.g., [1, 2, 3, 4, 5]). KGQA systems are dedicated to bridging the gap between Linked Data and end-users by converting natural-language (NL) questions into structured queries (e.g., SPARQL queries). Multilingual KGQA aims to retrieve answers from a KG for questions in multiple languages. During answer generation for a question, a (monolingual or multilingual) KGQA system usually creates a ranked list of

4TH INTERNATIONAL WORKSHOP ON KNOWLEDGE GRAPH GENERATION FROM TEXT (TEXT2KG) Co-located with the Extended Semantic Web Conference (ESWC 2025)

 alexander.gashkov@gmail.com (A. Gashkov); maria.eltsova@gmail.com (M. Eltsova);

aleksandr.perevalov@htwk-leipzig.de (A. Perevalov); andreas.both@htwk-leipzig.de (A. Both)

 0000-0001-6894-2094 (A. Gashkov); 0000-0003-3792-8518 (M. Eltsova); 0000-0002-6803-3357 (A. Perevalov); 0000-0002-9177-5463 (A. Both)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

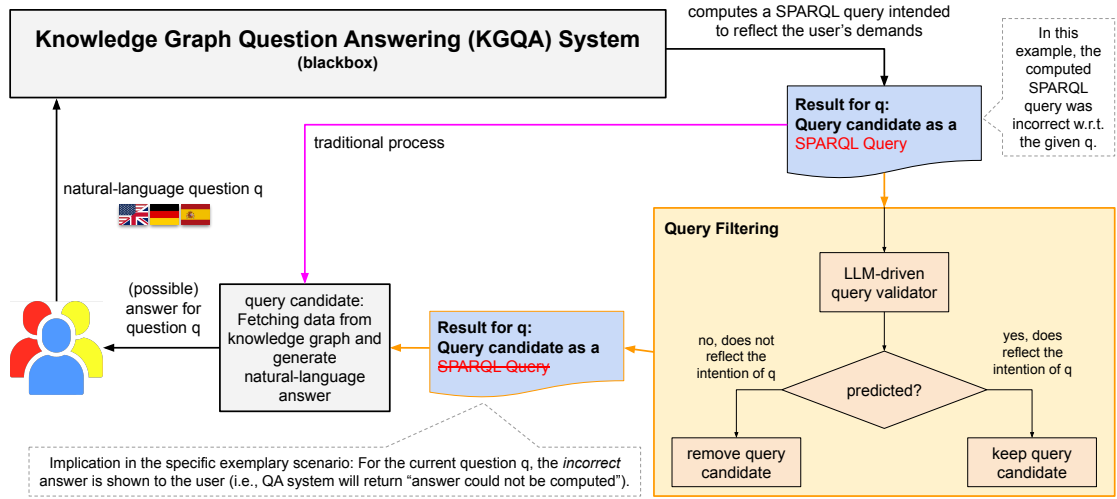


Figure 1: Big Picture: Query candidate filtering of multilingual KGQA systems.

SPARQL queries that are (hopefully) suitable for retrieving the correct answers to a particular question from a knowledge graph. Thereafter, a ranked Top- N of the retrieved answers becomes visible to the end-users (usually N is 1). The KGQA is supposed to always generate a correct answer based on the information from the deployed KG. However, even the best real-world KGQA systems often provide erroneous answers: according to Zimina et al. [6], Precision of the analyzed systems varies from 0.22 to 0.66. Hence, there is still the actual challenge that some of the *incorrect queries still could be prioritized over the correct ones*. This leads to a decrease in the quality and, therefore, the trustworthiness of a KGQA system. However, research often forgets the fact that trustworthiness is important, especially if the KGQA quality is not high enough, which is typically the case for non-English KGQA systems. Therefore, our approach aims at removing incorrect SPARQL queries, such that, in the worst case, the user is presented with no answer rather than the wrong one.

This paper tackles the mentioned challenge by introducing a SPARQL query filtering approach that uses LLMs to differentiate between correct and incorrect SPARQL queries. The approach is language- and KG-agnostic and can significantly improve the results of any KGQA system. If the system generates a list of at least two query candidates, the improvement by using our approach can be more considerable due to re-ranking the candidates in the list. The general idea of the approach introduced by this paper is presented in Figure 1.

Therefore, our research is aimed at answering the following research questions:

RQ1: To what extent is it possible to provide a generalized validation process for SPARQL queries that increases the quality and trustworthiness of the answers of a KGQA system?

RQ2: How can we create a language- and KGQA-agnostic validation process?

RQ3: What is the possible best result using state-of-art LLMs?

These research questions are intended to highlight the possibility of using LLMs for the task of SPARQL query validation based on a NL question.

Our approach was evaluated on the well-known multilingual QALD-9-plus [7, 8] dataset

(English, German, and Spanish languages) and two real KGQA systems (QAnswer [9] and MST5 [10]). The experiments were conducted by utilizing various LLMs of different sizes ranging from 7B to 123B parameters. Obtained results show a strong impact on the quality regarding both scores and all languages.

This paper has the following structure. First, we describe the related work (see Section 2) followed by the presentation of our approach in Section 3. Section 4 highlights the used QA system, dataset, and LLMs and describes the setup and execution of our experiments, whose data are evaluated and analyzed in Section 5. Section 6 briefly describes limitations and discusses the results. Section 7 concludes the paper and outlines future work. The data is available in the online appendix at: <https://github.com/WSE-research/Validation-2025-Data>.

2. Related Work

QA is one of the most important research fields in natural language processing (NLP). With the advent of LLMs, the research interest in the QA problems has been increasing. Currently, there are many papers covering the benefits of exploiting the LLMs for many KGQA tasks (e.g., [2, 4, 5, 11, 12], etc.). However, the problem of the **multilingualism in the field of KGQA** remains still underinvestigated but very important for both researchers and users. Most research in the area of KGQA is still focused on monolingual (i.e., English) settings since both building a large-scale KG and annotating QA data is expensive for each new language. Hence, our search indicates that multilingualism in KGQA is still a major challenge due to, on the one hand, the saturation of the KGQA field with work on English data (the inherent challenges of translating datasets and the reliance on English-only knowledge bases) [13] and, on the other hand, the scarcity of both the multilingual KGQA systems [14, 15, 16] and multilingual datasets [13, 14, 15, 17, 18, 19, 20, 21]. However, recently, there has been a rising demand for multilingual QA systems, which motivates researchers to focus on the problem of multilingual QA. To bridge this gap, many multilingual solutions for QA use machine translation (MT) for translating input questions (e.g., [22, 23]), which can be easily integrated into a monolingual system. However, this way, it highly depends on the quality of the used MT methods and is not able to provide users with a good quality [21, 22] due to the limitation of a small set of languages covered by existing KGs [16].

Other solutions utilize cross-lingual knowledge transfer or implement multilingual LMs (e.g., [5, 24, 25, 26]). Despite the promising way to cover a lack of multilingual data, these approaches do not always produce acceptable results (e.g., increase of F1 score by 0-7%), as this can incur the risk of negative transfer when there exists a large language shift [27]. Another problem we faced when analyzing the contribution of other researchers is that the data provided by them could not be compared properly due to different focuses, languages, metrics used, etc.

The **problem of query validation** is also a novel one in the field of KGQA. Query validation is understood as the process of checking the validity of the provided query with respect to the asked question, which can improve both the quality and performance of a QA system, being beneficial for knowledge-intensive and expert-reliant tasks that require evidence to validate generated text outputs. However, there is just a very limited number of studies on answer and/or query validation in the context of KGQA systems (e.g., [15, 28]). On the other hand,

there appeared recently some approaches to semantic parsing by treating it as a problem of semantic graph generation and re-ranking [29, 30, 31]. The recent implementation of LLMs in similar tasks [32, 33, 34, 35] could be a promising direction for enhancing the answer (or query) validation systems.

3. Approach

Our approach deals with filtering incorrect SPARQL query candidates generated by a KGQA system in response to a natural-language question. Questions are considered in multiple languages, which generalizes our approach more. Our approach’s core is to employ instruction-tuned LLMs for binary classification tasks as filters eliminating incorrect SPARQL queries. In this work, we tackle the problem of query validation considering a KGQA system as a black-box where the input is a question and the output is an answer in a natural-language form (cf. Figure 1). It is providing a “user” with one answer (top-ranked candidate) and cannot affect a KGQA system in any way. Hence, *we do not evaluate the quality of an entire QA system, but only the quality of a query validation module.*

Let QAS represent a KGQA system, s.t., $QAS^q : NL_q \rightarrow C_q$, where:

- Input: NL_q denotes a natural-language question written in a specific language (e.g., German), where q represents an identifier of the question in a dataset.
- Output: $C_q = \{SPARQL_1, SPARQL_2, \dots, SPARQL_k\}$ represents the output of the KGQA system for the question NL_q . C_q is an ordered collection (i.e., list) of SPARQL query candidates, which may be (1) empty, (2) contain one or multiple correct queries, or (3) consist entirely of incorrect queries.

Each question q has a list of *ground truth answers* \mathcal{A} defined by a dataset (can be empty). Afterward, a SPARQL query produced by a QAS returns another list of answers \mathcal{A}' as *predicted*. Therefore, we evaluate *correctness* of a query with a function $isCorrect$ that (1) takes answers generated by a $SPARQL_i$ query \mathcal{A}'_i and the ground truth answers \mathcal{A}_i as input, (2) calculates the F1 score over the provided answer sets, and (3) assigns a *label* = {*correct*, *incorrect*} that indicates the correctness of the answer of this query as follows:

$$isCorrect(\mathcal{A}_i, \mathcal{A}'_i) = \begin{cases} correct, & \text{if F1 score}(\mathcal{A}_i, \mathcal{A}'_i) = 1.0 \\ incorrect, & otherwise \end{cases} \quad (1)$$

Therefore, to enhance the QA quality by filtering SPARQL query candidates, we need to create a function F that represents a binary classifier, s.t., $F : (NL_i, SPARQL_i) \rightarrow label$. Since the filtering function F *does not reorder the list but eliminates list items marked as incorrect*, the correct query can only be placed at the top of the list if all incorrect ones before it are removed.

Verbalization and Binary Classification of SPARQL Queries. To create the filtering function F , we exploit LLMs (cf. Section 4.1). Many KGs do not provide human-readable URIs of their entities (e.g., Abraham Lincoln is denoted as Q91¹ in Wikidata), therefore, we suppose that SPARQL queries for such KGs should be verbalized, i.e., transformed to a NL-like representations while using labels of the corresponding entities from a given KG (e.g., Wikidata).

¹<https://www.wikidata.org/wiki/Q91>

Review the provided SPARQL query and the question.
The query:
SELECT DISTINCT ?s1
WHERE {
? s1 ?p1 wd:Q571 .
? s1 wdt:P50 wd:Q2331679 .
}
The question: Wer ist der Autor des Buches Traumdeutung?
The labels in the query are:
wd:Q571 - Buch,
wdt:P50 - Autor,
wd:Q2331679 - Stanley Deser.
Are the query and the question identical? Answer Yes or No.

Figure 2: Prompt with the knowledge injection. An example of an **incorrect** SPARQL for the German question “Wer ist der Autor des Buches Traumdeutung?” (English: “Who is the author of the interpretation of dreams?”).

Review the provided SPARQL query and the question.
The query:
SELECT DISTINCT ?uri WHERE { wd:Q319308 wdt:P166 ?uri . }
Which awards did Douglas Hofstadter win?
The labels in the query are:
wd:Q319308 - Douglas Hofstadtesr,
wdt:P166 - award received
Are the query and the question identical? Answer Yes or No.

Figure 3: Prompt with the knowledge injection. An example of a **correct** SPARQL for the English question “Which awards did Douglas Hofstadter win?”).

For example, a NL question *What country is Mount Everest in?* has a following SPARQL representation `SELECT DISTINCT ?o1 WHERE { wd:Q513 wdt:P17 ?o1 . }` and a following low-level verbalization *The query is: SELECT DISTINCT ?o1 WHERE { wd:Q513 wdt:P17 ?o1 . }*. *The labels in the query are: wd:Q513 - Mount Everest, wdt:P17 - country.*

Knowledge injection provided with a prompt grants information to LLMs about the textual representation of the URI.

Evaluation of QA Quality. For measuring the effect of our approach (e.g., the SPARQL query filtering) on QA quality, we use the relative metrics of answers quality which are calculated based on the *ATS* and *Recall* before and after applying the approach (in this paper, we are taking into consideration only top-1 query). It is worth mentioning that the QALD-9-plus benchmark is supposed to have at least one correct answer for each question, and each top-1 SPARQL query generated by KGQA is treated as predicted correctly. In this particular case, $P@1$ and $R@1$ are always the same and equal to the $F1@1$ score.

We use the Answer Trustworthiness Score ATS to estimate trustworthiness of QA system (following the definition in [20]), where for all questions q in a dataset D_i , a score per question is computed, summed up, and normalized in the range of -1 to $+1$. Following the statement “no answer is better than wrong answer”, there is no penalty if a KGQA system returns no result (i.e., systems showing fewer incorrect answers to users achieve a higher score).

The QA system can certainly achieve the average ATS of 0 just by responding with no answer to all questions in D . To achieve the positive ATS , a QA system must provide more correct than incorrect answers (cf. Figure 5). Thus, the ATS is more strict than other common metrics and an ideal metric for measuring the quality of KGQA systems. In this paper, we use $rATS$, the relative score, to measure the impact of the validation process.

The second metric, relative recall (rR), shows how many correct answers were removed from the answers pool. It counts from 0.0 to 1.0; the higher the metric is, the better quality the validator has. The value of 0.0 means that all the right answers were removed from the answers pool (cf. Figure 4).

As mentioned above, in this particular case, all the metrics – Precision, Recall, and F1 – are equal, therefore, we do not need to calculate Precision and F1.

Quality and Validation process. It is obvious that the answers’ quality after validation is strictly dependent on the quality of KGQA results.

The baseline for the ATS is its value calculated for the QA before validation, treated also as lower bound ATS_{lo} . The highest bound for the ATS (or maximal achievable value) is calculated for a perfect process outcome: all incorrect answers are removed, all correct are preserved (cf. Figure 4, Figure 5). If QA system produces N_c correct answers and N_i incorrect, then bounds can be calculated with the following formulas.

$$ATS_{lo} = \frac{N_c - N_i}{N_c + N_i}$$

$$ATS_{hi} = \frac{N_c}{N_c + N_i}$$

After defining the bounds, we can set a new metric: relative ATS change. Let ATS' be the ATS of QAS after validation. Then, the relative ATS change can be easily found as:

$$rATS = \frac{ATS' - ATS_{lo}}{ATS_{hi} - ATS_{lo}}$$

$rATS$ is less dependent on the quality of KGQA and, thus, more robust and informative.

4. Experimental Setup

4.1. Material

In this part, we briefly describe the components used to manifest the experimental environment: QA systems, Dataset, and LLMs.

The KGQA systems QAnswer and MST5. Out of many existing QA systems, we have chosen a state-of-the-art QAnswer because of its multilingualism (the system allows the use of 8

languages), support for multiple KGs (including Wikidata), robustness (cf. [36]), portability, and accessibility (cf. [9]). Additionally, it demonstrates high precision and recall, e.g., a high answer quality [20, 37, 36, 38, 39], as well as it provides an API for asking a question and receiving the corresponding ranked query candidate list (up to 60 candidates) [36].

MST5 presents a new strategy for multilingual KGQA. It emphasizes incorporating and utilizing additional knowledge, such as entity link tags and linguistic context, via a transformer-based model [10]. The MST5 approach proposes that linguistic context and entity information is extracted from the input NL question. Then, the extracted information is concatenated with the input before being passed on to the language model. The language model generates the resulting SPARQL query. MST5 significantly outperforms the competing systems (DeepPavlov-2023 [40], QAnswer [36, 41], etc.) on all supported languages but also achieves comparable results on most supported languages except the low-resource languages [10].

QALD-9-plus Dataset. The scarcity of datasets for KGQA, especially multilingual benchmarks, is a crucial problem in the field, indicated in recent research (e.g., [10, 13, 14, 17, 18, 19, 20, 21], etc.). The QALD datasets represent a series of well-established benchmarks for multilingual KGQA. QALD-9 [17] consists of 558 questions accompanied by a textual representation in multiple languages, the corresponding SPARQL query (over DBpedia), the answer entity URI, and the answer type. *QALD-9-plus*² [7] is an extension of the QALD-9 dataset where Spanish was added via [8], and the translation quality for existing languages was significantly optimized by validations of native speakers. Therefore, the dataset supports English, German, Russian, French, Spanish, Armenian, Belarusian, Lithuanian, Bashkir, and Ukrainian. Moreover, QALD-9-plus also supports the Wikidata knowledge graph.

Other multilingual datasets – RuBQ 2.0 [38], MCWQ [19], and Mintaka [42] – have some flaws and restrictions, e.g., (1) missing ground truth (Mintaka), (2) another set of languages or few languages (RuBQ 2.0 and MCWQ), (3) machine-translated questions without any post-editing (RuBQ 2.0), etc. These drawbacks do not allow us to use them in our experiments because of the non-comparability of the data.

LLMs. We used LLMs of five different publishers: Open AI, DeepSeek, Qwen, Mistral, and Llama.

OpenAI’s LLMs (e.g., **GPT-4**) represents a significant advancement in the field of AI, offering substantial improvements over its predecessors in terms of multimodal capabilities of processing image and text inputs and producing text outputs, context window size, tokenization efficiency, and processing speed [43]. This is a transformer-based model pre-trained to predict the next token in a document. The model **o1-mini** is, according to its developers (Open AI)³, a cost-efficient reasoning model that excels at STEM, especially math and coding. Both OpenAI models are multilingual.

Qwen 2 series released in 2024 is a versatile suite of foundational and instruction-tuned language models, ranging from 0.5 to 72 billion parameters [44]. Qwen 2.5 is grounded in the transformer architecture and trained using next-token prediction. During the experiments and evaluation, the LM showed robust multilingual capabilities and was proficient in approximately 30 languages, including English, Spanish, and German.

²https://github.com/KGQA/QALD_9_plus

³<https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/>

Mistral Small 3⁴ is a pre-trained and instructed model catered to those of generative AI tasks that require robust language and instruction following performance. According to the developers, this new model was designed to saturate performance at a size suitable for local deployment. Particularly, Mistral Small 3 has far fewer layers than competing models, substantially reducing the time per forward pass.

Mistral Large⁵ is a new cutting-edge text generation model that can be used for complex multilingual reasoning tasks including “text understanding”. The developers claim that it is natively fluent in English, French, Spanish, German, and Italian, with a nuanced understanding of grammar and cultural context.

The Meta **Llama 3.3**⁶ multilingual large language model (LLM) is an instruction-tuned generative model in 70B (text in/text out). The developers pointed out that the Llama 3.3 instruction-tuned text-only model is optimized for multilingual dialogue use cases. It is an auto-regressive LM that uses an optimized transformer architecture.

DeepSeek-R1 [45] incorporates multi-stage training and cold-start data before reinforcement learning. According to its developers, DeepSeek-R1 is currently optimized for Chinese and English, which may result in language mixing issues when handling queries in other languages. The developers also observed that the model is sensitive to prompts and, therefore, advise users to directly describe the problem and to specify the output format using a zero-shot setting for optimal results.

4.2. Experimental Design

In our experiments, we used two sets of data, each obtained as the first SPARQL query given respectively by QAnswer and MST5 in answer for each QALD-9-plus question (including both test and train splits) in three languages: English, German, and Spanish. The first set obtained by QAnswer consists of 130 correct and 308 incorrect queries for English, 104 correct and 461 incorrect queries for German, and 65 correct and 375 incorrect queries for Spanish. MST5 provided us with a set of 81 correct and 90 incorrect queries for English, 73 correct and 125 incorrect queries for German, and 81 correct and 135 incorrect queries for Spanish. QAnswer produced more queries, both correct and incorrect, than MST5, moreover, the set of incorrect queries is nearly three to five times larger than the set of correct ones.

In step one S_1 , all queries were evaluated using Equation 1 to find the initial quality metrics of both KGQA. Then, we formed a prompt (cf. Figure 2) with the knowledge injection, sent it to all LLMs involved in the experiments, and evaluated the metrics after filtering.

In the next step, S_2 , we performed the validation itself and evaluated quality after query filtering and validation effectiveness.

As described in Section 4.1, we use five groups of LLMs, namely the model of OpenAI, DeepSeek, Qwen, Llama, and Mistral. The detailed experimental setup for S_1 and S_2 is described in the following subsections.

⁴<https://mistral.ai/en/news/mistral-small-3>

⁵<https://mistral.ai/news/mistral-large>

⁶<https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct>

4.3. S_1 – Query Evaluation

At this step, we execute all ground truth SPARQL queries from QALD and queries produced by both KGQA systems on Wikidata to get the answer sets. SPARQL query returns a number of rows, one for each match. Match can be an entity, a predicate, a literal, or a set of entities and/or predicates and/or literals. The sets returned by the gold standard query and the candidate must exact the same to evaluate the candidate as correct. As all the QALD questions have a non-empty answer set, all candidates, both correct and incorrect, contribute to the metric calculation.

The models from all groups were taken “as-is” and were instructed with zero-shot prompts that use the knowledge injection technique. The prompts contain a NL_q , a raw $SPARQL_q$ and a $(URI, label)$ tuples, which is a knowledge injection part retrieved from Wikidata (see Figure 2). Based on the aforementioned information, the models are instructed to generate “yes” or “no”, corresponding to a correct or incorrect result. The temperature parameter was set to 0, where possible, and the other parameters were kept with default values.

The GPT models were used via the official OpenAI Python library⁷. Other models were executed on a local server powered by two NVIDIA L40S GPUs (48GB VRAM).

4.4. S_2 – Query Filtering

For the evaluation of the effect of SPARQL query filtering, we calculate such metrics as $rR@1$ and relative change of Answer Trustworthiness Score ($rATS@1$). First, we do not estimate the quality of the QA systems while evaluating only the quality of query validation. Therefore, the traditional metrics, like Precision, Recall, F1 score, etc., are not applicable. Second, the idea of most QA systems is that they provide the user with only one answer, usually generated from a top SPARQL query in a ranked list. Hence, the main task of the classifier was to filter out the incorrect answers. Third, since MST5 provides only one query candidate, we can use metrics for only the top candidate (@1) to properly compare the applicability of our approach to two different QA systems.

We used three languages, both presented in the dataset and supported by QAnswer and MST5 (English, German, and Spanish), which have enough data for experiments. Both systems provide good-quality queries.

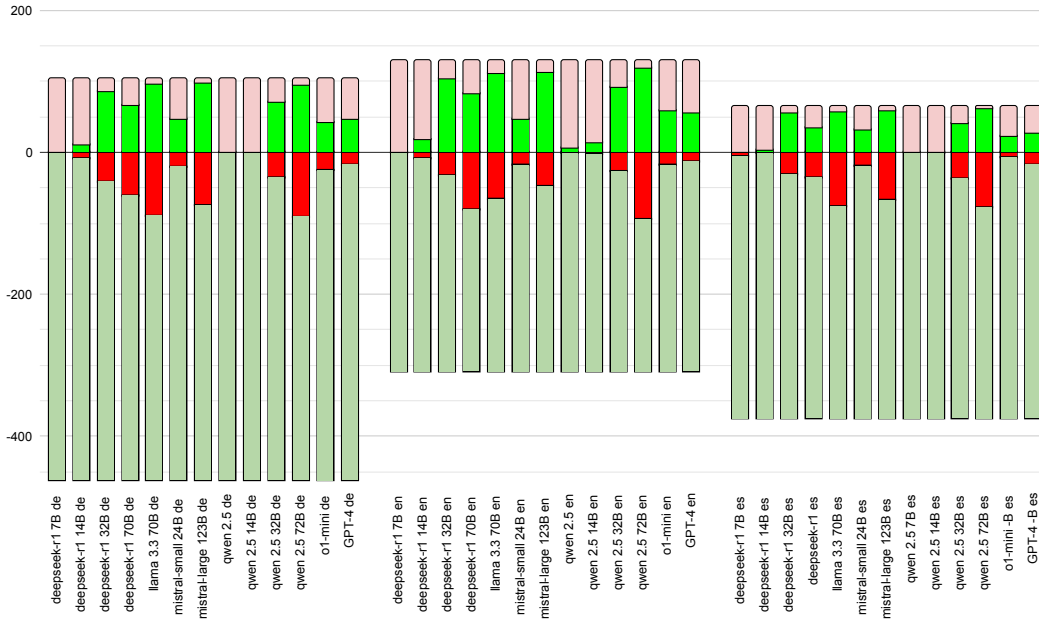
Unfortunately, we cannot automatically evaluate the semantics of a SPARQL query, so we consider all semantic flaws leading to no response from Wikidata as unrecoverable errors.

Query filtering was done as follows. If LLM answered “no” to the question “Are the question and the query identical?”, the query was removed, and the list of queries became empty because we took into consideration only the top one SPARQL query. If LLM answered “yes”, the query was kept. For filtering, we used the same procedures as for the evaluation before.

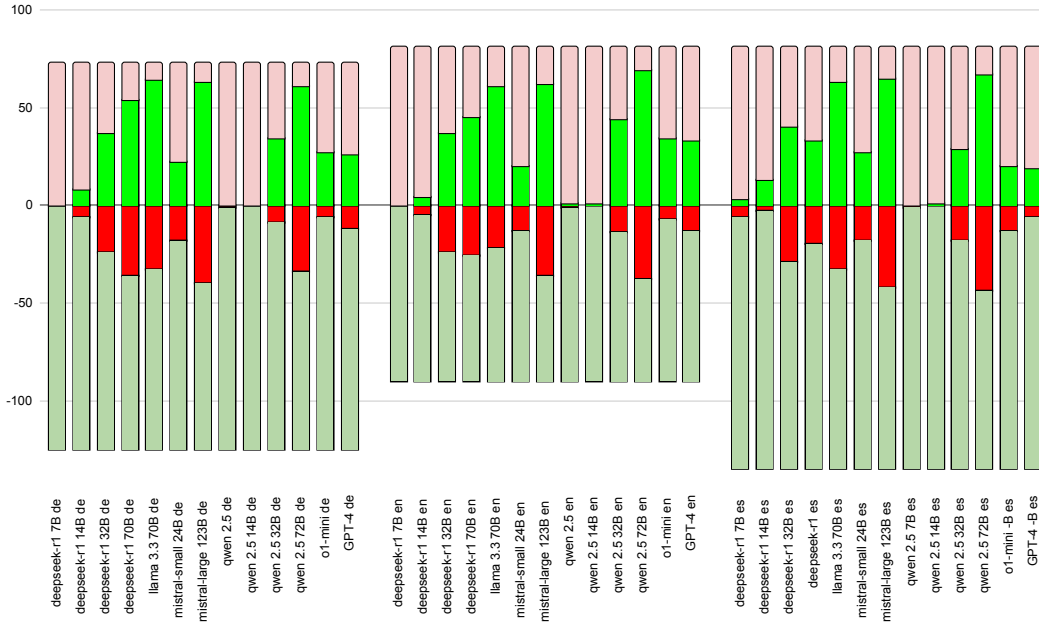
5. Evaluation and Analysis

Table 1 contains the experimental results before and after the query validation process for all LLMs and QA systems executed on three languages. The columns use the following symbols

⁷<https://github.com/openai/openai-python>



(a) Results for QAnswer: Number of preserved correct and removed incorrect query candidates.



(b) Results for MST5: Number of preserved correct and removed incorrect query candidates.

Figure 4: Overview of the models' performance grouped by languages. Each complete bar represents the number of questions answered by the corresponding QA system (the negative values count incorrect query candidates, the positive values – correct query candidates in the original answer of the systems). The colors mean: **the number of correct queries preserved during the filtering (as intended)**, **correct queries filtered out (not intended)**; **number of incorrect queries preserved after filtering (not intended)**, **incorrect candidates filtered out (as intended)**. Therefore, the more correct query candidates are preserved, the better is the result (in the ideal case, there is a complete solid green bar); the more incorrect query candidates are removed (a perfect result: all incorrect queries would be removed, indicated by a full light green bar below the zero line), the better is the performance of the query validator.

(legend): N_c , N_i – number of correct and incorrect answers before filtering, respectively; N'_c , N'_i – number of correct and incorrect answers after filtering, respectively; $rATS@1$, $rR@1$ – relative change of *Recall*@1 and *ATS*@1 in the validation process. The statistics of the validation process are demonstrated in Figure 4. We determine the best-performing model while aiming at $rATS@1$ and $rR@1$.

As the *ATS* reflects the idea of “no answer is better than a wrong answer”, the results after filtering demonstrate huge improvements, showing that our approach has a very strong impact on the QA trustworthiness given the reference QAnswer and MST5 systems. Both Table 1 and Figure 4 demonstrate that the LLMs of the smallest size (all models of 7B and 14B parameters) tend to estimate all candidates as incorrect and, therefore, eliminate them. In this case, the $ATS=0$ after the filtering, so the $rATS@1$ is not very high, moreover, the $rR@1$ usually equals 0 or is slightly above 0, i.e., users always get from a QA system an answer like “Sorry, the correct answer could not be computed”, which could not suit them.

The larger LLMs of all groups demonstrate much better improvements regarding both metrics, however, they tend to preserve nearly all correct queries while also keeping many incorrect ones. Therefore, demonstrating a pretty high value of $rATS@1$, they are losing in $rR@1$. Another obvious thing here is that there were preserved less correct answers for English in contrast to German and Spanish. Moreover, the $rATS@1$ for English is also lower than for German and Spanish. The reason for this phenomenon could be the fact that there were fewer incorrect and more correct queries before filtering for English, i.e., the initial quality of QA systems (without validation) is higher for English than for other languages. Therefore, implementing our validation approach into a QA system can also significantly improve its non-English output.

Regarding the $rATS@1$ metrics, the best improvement demonstrates GPT-o1-mini, however, the model preserves less than 50% of the correct answers (46.2% for English, 41.3% for German, and 36.9% for Spanish on QAnswer, the values on MST5 are lower; the value of $rR@1$ demonstrates these facts).

Regarding both metrics, Llama 3.3, Mistral Large 3, and Qwen 2.5 (72B) demonstrate the best improvement for all languages. However, there is no LLM at the moment which is close to an ideal result: all correct queries are preserved, and all incorrect ones are filtered out. We should also point out that, according to our results, the implementation of our approach currently grants more benefits to QAnswer.

Figure 5 illustrates the nature of $ATS@1$. On the results obtained with all models of Qwen 2.5⁸ for both QA systems and all three languages, this figure presents the lowest value ($ATS@1_{lo}$), the achieved value ($ATS@1$), and maximal value ($ATS@1_{hi}$). We chose these models for illustration because of their four different sizes and their demonstration of the main trends described above. In other words, this graphic represents the relative improvement of a QA system exploiting our approach. According to results presented in Table 1 and by Figure 5, our approach could improve a QA system when applying a LLM of any publisher and size: even the smaller models (7B and 14B) grants an increase of *ATS* for all languages on the both exploited state-of-the-art QA systems while the larger models provide much higher quality in terms of both metrics. Our approach is also able to provide the $ATS@1$ close to the maximal value of it

⁸The chart with all data for all models could be found in our online appendix <https://github.com/WSE-research/Validation-2025-Data>.

Table 1
Results of the validation process

Model	Parameters	Language	QAnswer						MST5					
			N_i	N_c	N'_i	N'_c	$rATS@1$	$rR@1$	N_i	N_c	N'_i	N'_c	$rATS@1$	$rR@1$
DeepSeek-R1	7B	en	308	130	1	0	0.575	0.000	90	81	0	0	0.100	0.000
	7B	de	461	104	0	0	0.774	0.000	125	73	0	0	0.416	0.000
	7B	es	375	65	5	0	0.813	0.000	135	81	6	3	0.378	0.037
	14B	en	308	130	8	19	0.614	0.146	90	81	5	4	0.089	0.049
	14B	de	461	104	9	11	0.779	0.106	125	73	6	8	0.432	0.110
	14B	es	375	65	1	3	0.832	0.046	135	81	3	13	0.474	0.160
	32B	en	308	130	33	104	0.808	0.800	90	81	24	37	0.244	0.457
	32B	de	461	104	42	86	0.870	0.827	125	73	24	37	0.520	0.507
	32B	es	375	65	31	56	0.893	0.862	135	81	29	40	0.481	0.494
	70B	en	308	130	80	84	0.591	0.646	90	81	26	45	0.311	0.556
	70B	de	461	104	61	67	0.787	0.644	125	73	36	54	0.560	0.740
	70B	es	375	65	35	36	0.829	0.554	135	81	20	33	0.496	0.407
Llama 3.3	70B	en	308	130	66	112	0.727	0.862	90	81	22	61	0.533	0.753
	70B	de	461	104	90	97	0.790	0.933	125	73	33	64	0.664	0.877
	70B	es	375	65	76	58	0.779	0.892	135	81	33	63	0.622	0.778
Mistral Small 3	24B	en	308	130	18	48	0.675	0.369	90	81	13	20	0.178	0.247
	24B	de	461	104	21	48	0.833	0.462	125	73	18	22	0.448	0.301
	24B	es	375	65	19	33	0.864	0.508	135	81	18	27	0.467	0.333
Mistral Large 3	123B	en	308	130	47	113	0.792	0.869	90	81	36	62	0.389	0.765
	123B	de	461	104	74	99	0.829	0.952	125	73	40	63	0.600	0.863
	123B	es	375	65	67	60	0.808	0.923	135	81	42	65	0.570	0.802
Qwen 2.5	7B	en	308	130	1	7	0.597	0.054	90	81	1	1	0.100	0.012
	7B	de	461	104	0	0	0.774	0.000	125	73	1	0	0.408	0.000
	7B	es	375	65	0	0	0.827	0.000	135	81	1	0	0.393	0.000
	14B	en	308	130	2	14	0.617	0.108	90	81	0	1	0.111	0.012
	14B	de	461	104	0	1	0.777	0.010	125	73	0	0	0.416	0.000
	14B	es	375	65	1	1	0.827	0.015	135	81	0	1	0.407	0.012
	32B	en	308	130	27	93	0.792	0.715	90	81	14	44	0.433	0.543
	32B	de	461	104	35	71	0.852	0.683	125	73	9	34	0.616	0.466
	32B	es	375	65	37	41	0.837	0.631	135	81	18	29	0.481	0.358
	72B	en	308	130	94	119	0.659	0.915	90	81	38	69	0.444	0.852
	72B	de	461	104	91	96	0.785	0.923	125	73	34	61	0.632	0.836
	72B	es	375	65	78	62	0.784	0.954	135	81	44	67	0.570	0.827
Open AI o1-mini	-	en	308	130	17	60	0.718	0.462	90	81	7	34	0.400	0.420
	-	de	461	104	25	43	0.813	0.413	125	73	6	27	0.584	0.370
	-	es	375	65	7	24	0.872	0.369	135	81	13	20	0.452	0.247
Open AI GPT-4	-	en	308	130	13	57	0.721	0.438	90	81	13	33	0.322	0.407
	-	de	461	104	18	47	0.837	0.452	125	73	12	26	0.528	0.356
	-	es	375	65	17	28	0.856	0.431	135	81	6	19	0.496	0.235

(e.g., with Qwen 2.5 with 32B parameters for English on QAnswer and German on MST5, cf. Figures 5a and 5d).

6. Limitations and Discussion

In this paper, our attention was concentrated on the ability of LLMs to play the role of “query validator”, i.e., to distinguish between correct and incorrect query candidates generated by a KGQA system from a natural-language question. Before discussing the results, we intend to point out some limitations of this research. First of all, we utilized only one dataset, QALD-9-plus, because of the scarcity of good-quality multilingual benchmarks (Sections 2 and 4.1). The second limitation was the use of only two modern QA systems, realizing different strategies of answering questions over KG. Another limitation may concern the choice of only three frequently used institutional languages from the Indo-European language family. Finally, we used for this research only one variant of prompt and leave the exploration of this direction for future work. However, being language-agnostic, portable, robust, and easy reproducible, our approach provides a wide field to investigate all the limitations in future research, e.g., the experiments could be reproduced on other benchmarks (datasets, QA systems, LLMs) and, therefore, other languages; different type of prompts (language-specific, LLM-specific, multi-shot, etc.) may be used to identify the best solution for each case.

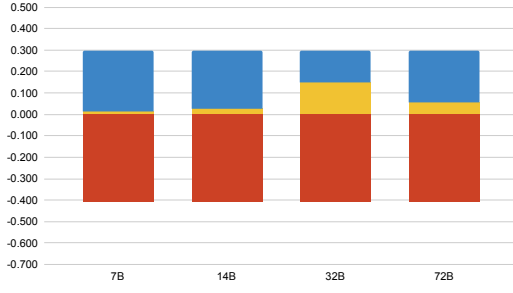
In this study, we relayed on gold standard queries (ground-truth) only to evaluate the quality of the validation. The modals used as validators do not depend on ground-truth.

Our results prove that our approach has a strong impact on the validation quality in all three languages. All LLMs demonstrate significant improvement w.r.t. both metrics used. While the smallest models (7B parameters) show the trend to filter out all candidates, i.e., the *ATS* was under 0 before the filtering and equals 0 after it, most of the larger models are able to filter out more incorrect candidates by preserving the correct ones. Post-experiment analysis has shown that the integration of larger LLMs into our approach further improves the overall quality of the QA systems. These observations highlight a crucial problem concerning the size of LLMs: the larger LLMs provide better output, however, they require more and more computing resources. But are these costs correlated with the obtained results, and do they really provide the expected quality improvement? The answers to these questions might be a problem of special research. However, every researcher can decide which LLM to use for the specific research aims.

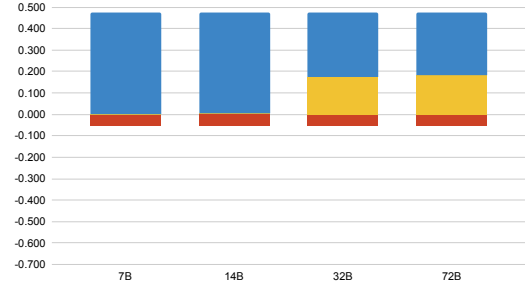
7. Conclusions and Future Work

In this paper, we presented an easy-to-realize but effective approach for improving the quality of Question Answering over Knowledge Graphs. In particular, our approach is able to remove incorrect query candidates, s.t., the number of incorrect results shown to the users is significantly reduced – an argument that strongly enhances the trustworthiness of QA systems. Moreover, we concentrate our work on developing an approach that also applies to non-English questions without using machine translation. Summing up, the unique features of our approach are as follows:

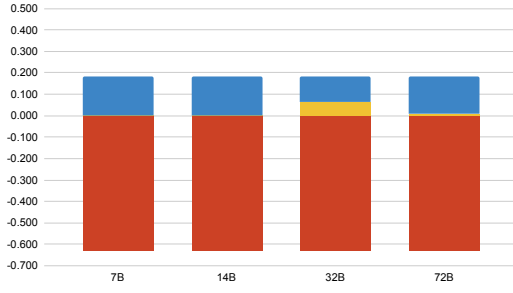
- (1) The system-agnostic decision, which is built on top of the query candidates represented, utilizes the SPARQL format as it is typical in the field of KGQA (answer to the **RQ1**). Hence,



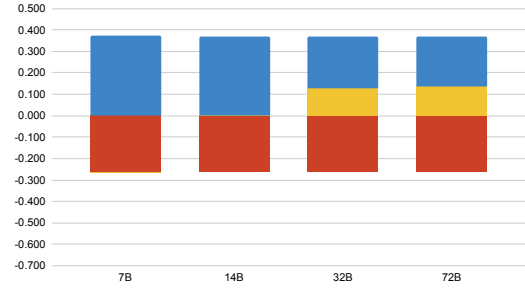
(a) QAnswer, English



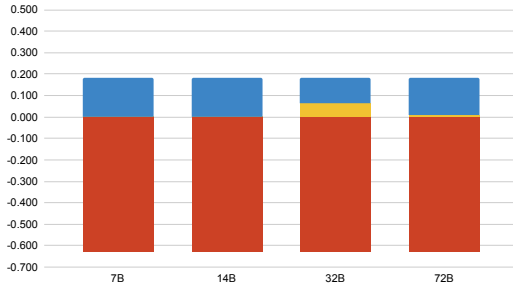
(b) MST5, English



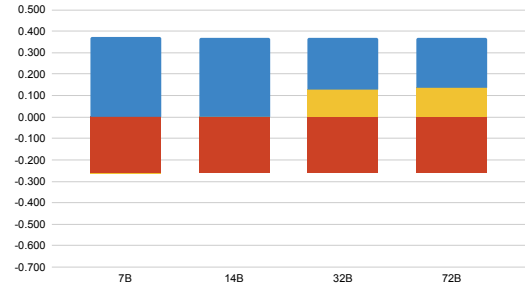
(c) QAnswer, German



(d) MST5, German



(e) QAnswer, Spanish



(f) MST5, Spanish

Figure 5: Example of dynamics of ATS@1 metrics for English, German, and Spanish on QAnswer and MST5 for all models of Qwen 2.5. The **red bars** illustrate the ATS@1 before filtering, which varies for different languages and QA systems. The **blue bars** show the maximal value of ATS@1 for each language and each QA system. The **yellow bars** demonstrate ATS@1 after filtering. In the ideal case, the yellow bar equals the blue one (i.e., it would not be visible anymore).

our approach can be implemented into any existing QA system to improve its answer quality (i.e., its trustworthiness).

(2) We created a language-agnostic approach. Hence, it can be transferred to other languages without changing the process itself. The only requirement is the representation of language-specific labels in the considered Knowledge Graph. Obtained results demonstrate that our

approach is applicable to other languages and will improve the quality of questions represented in other languages as well, and with a higher increase of trustworthiness as for English (answer to the **RQ2**).

(3) All LLMs, both larger and smaller, can be exploited for our approach, so that users have the choice of the technology being used. Our experiments show a strong quality improvement for all the LLMs families we used for our research; besides, the larger models (32B parameters or more) demonstrated more impressive results. However, their implementation might signify a much higher computational time investment and/or cost-per-interaction (answer to **RQ3**). In this research, we did not observe an advantage of exploiting the commercial LLMs over the open-source LLMs.

Future work might deal with experiments both with a language-specific prompt and an LLM-specific prompt. Our approach could also be extended by using additional KG properties. Moreover, an interesting direction would be a combined usage of LLMs by generating SPARQL queries from NL questions and validating them (e.g., filtering out the incorrect ones). Furthermore, a promising direction to improve the question answering results for non-English systems would be to solve the problem of labels' non-availability for not frequently used or low-resource languages. Future studies will additionally include nDCG@k metrics (for a value of k , e.g., set to 5 or 10) to demonstrate more benefits of the proposed approach and its effectiveness in query validation and filtering strategies (like [15]). Additionally, measuring the impact in comparison to other quality-improving components (e.g., while integrating our approach as a component in KGQA frameworks like Qanary [46, 47, 48]) is a promising topic while aiming for balancing metrics like quality, costs, and runtime.

Declaration on Generative AI

During the preparation of this work, the authors used **Grammarly** to check grammar and spelling. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] J. Kocoń, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydło, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, et al., ChatGPT: Jack of all trades, master of none, *Information Fusion* 99 (2023) 101861.
- [2] R. Omar, O. Mangukiya, P. Kalnis, E. Mansour, ChatGPT versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots, *arXiv preprint arXiv:2302.06466* (2023).
- [3] S. Wang, H. Scells, B. Koopman, G. Zuccon, Can ChatGPT write a good boolean query for systematic review literature search?, in: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1426–1436.
- [4] Y. Tan, D. Min, Y. Li, W. Li, N. Hu, Y. Chen, G. Qi, Can ChatGPT replace traditional KBQA models? An in-depth analysis of the question answering performance of the GPT LLM family, in: *International Semantic Web Conference*, Springer, 2023, pp. 348–367.

- [5] N. Hu, J. Chen, Y. Wu, G. Qi, S. Bi, T. Wu, J. Z. Pan, Benchmarking large language models in complex question answering attribution using knowledge graphs, *arXiv preprint arXiv:2401.14640* (2024).
- [6] E. Zimina, K. Järvelin, J. Peltonen, A. Ranta, J. Nummenmaa, Traquila: Transparent Question Answering over RDF through linguistic analysis, in: *International Conference on Web Engineering*, Springer, 2024, pp. 19–33.
- [7] 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.
- [8] J. Soruco, D. Collarana, A. Both, R. Usbeck, QALD-9-ES: A Spanish Dataset for Question Answering Systems, *Studies on the Semantic Web*, IOS Press BV, 2023, pp. 38–52. doi:10.3233/SSW230004.
- [9] D. Diefenbach, A. Both, K. Singh, P. Maret, Towards a question answering system over the semantic web, *Semantic Web* 11 (2020) 421–439. doi:10.3233/SW-190343.
- [10] N. Srivastava, M. Ma, D. Vollmers, H. M. Zahera, D. Moussallem, A. N. Ngomo, MST5 - multilingual question answering over knowledge graphs, *CoRR abs/2407.06041* (2024). doi:10.48550/ARXIV.2407.06041. arXiv:2407.06041.
- [11] T. Mecharnia, M. d’Aquin, Performance and limitations of fine-tuned LLMs in SPARQL query generation, in: *Proceedings of the Workshop on Generative AI and Knowledge Graphs (GenAIK)*, 2025, pp. 69–77.
- [12] J. Lehmann, A. Meloni, E. Motta, F. Osborne, D. R. Recupero, A. A. Salatino, S. Vahdati, Large language models for scientific question answering: An extensive analysis of the SciQA benchmark, in: *European Semantic Web Conference*, Springer, 2024, pp. 199–217.
- [13] R. Cui, R. Aralikkatte, H. Lent, D. Hershcovich, Multilingual compositional Wikidata questions, *arXiv preprint arXiv:2108.03509* (2021).
- [14] A. Perevalov, A. Both, A.-C. N. Ngomo, Multilingual question answering systems for knowledge graphs—a survey, *Semantic Web* 15 (2024) 2089–2124. URL: <https://www.semantic-web-journal.net/system/files/swj3530.pdf>.
- [15] A. Perevalov, A. Gashkov, M. Eltsova, A. Both, Language models as SPARQL query filtering for improving the quality of multilingual question answering over knowledge graphs, in: K. Stefanidis, K. Systä, M. Matera, S. Heil, H. Kondylakis, E. Quintarelli (Eds.), *Web Engineering*, Springer Nature Switzerland, Cham, 2024, pp. 3–18.
- [16] L.-A. Kaffee, R. Biswas, C. M. Keet, E. K. Vakaj, G. de Melo, Multilingual knowledge graphs and low-resource languages: A review, *Transactions on Graph Data and Knowledge* 1 (2023) 10–1.
- [17] R. Usbeck, R. H. Gusmita, A.-C. N. Ngomo, M. Saleem, 9th challenge on question answering over linked data (QALD-9), in: *Semdeep/NLIWoD@ISWC*, 2018.
- [18] A. Saxena, S. Chakrabarti, P. Talukdar, Question answering over temporal knowledge graphs, *arXiv preprint arXiv:2106.01515* (2021).
- [19] R. Cui, R. Aralikkatte, H. Lent, D. Hershcovich, Compositional generalization in multilingual semantic parsing over Wikidata, *Transactions of the Association for Computational Linguistics* 10 (2022) 937–955.
- [20] A. Gashkov, A. Perevalov, M. Eltsova, A. Both, Improving question answering quality

through language feature-based SPARQL query candidate validation, volume 13261 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 217–235.

- [21] E. Loginova, S. Varanasi, G. Neumann, Towards end-to-end multilingual question answering, *Information Systems Frontiers* 23 (2021) 227–241. doi:10.1007/s10796-020-09996-1.
- [22] A. Perevalov, A. Both, D. Diefenbach, A.-C. Ngonga Ngomo, Can machine translation be a reasonable alternative for multilingual question answering systems over knowledge graphs?, in: *Proceedings of the ACM Web Conference 2022, WWW '22*, Association for Computing Machinery, 2022, p. 977–986. doi:10.1145/3485447.3511940.
- [23] N. Srivastava, A. Perevalov, D. Kuchelev, D. Moussallem, A.-C. Ngonga Ngomo, A. Both, Lingua franca – entity-aware machine translation approach for question answering over knowledge graphs, in: *Knowledge Capture Conference, K-CAP '23*, Association for Computing Machinery, 2023, p. 122–130. doi:10.1145/3587259.3627567.
- [24] Y. Zhou, X. Geng, T. Shen, W. Zhang, D. Jiang, Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021, pp. 5822–5834. doi:10.18653/v1/2021.naacl-main.465.
- [25] V. Konovalov, P. Gulyaev, A. Sorokin, Y. Kuratov, M. Burtsev, Exploring the BERT cross-lingual transfer for reading comprehension, in: *Computational Linguistics and Intellectual Technologies*, 2020, pp. 445–453.
- [26] T. Pires, How multilingual is multilingual BERT, arXiv preprint arXiv:1906.01502 (2019).
- [27] Z. Li, M. Kumar, W. Headden, B. Yin, Y. Wei, Y. Zhang, Q. Yang, Learn to cross-lingual transfer with meta graph learning across heterogeneous languages, in: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 2290–2301.
- [28] G. Maheshwari, P. Trivedi, D. Lukovnikov, N. Chakraborty, A. Fischer, J. Lehmann, Learning to rank query graphs for complex question answering over knowledge graphs, in: *International semantic web conference*, Springer, 2019, pp. 487–504.
- [29] S. W.-t. Yih, M.-W. Chang, X. He, J. Gao, Semantic parsing via staged query graph generation: Question answering with knowledge base, in: *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*, 2015.
- [30] M. Yu, W. Yin, K. S. Hasan, C. d. Santos, B. Xiang, B. Zhou, Improved neural relation detection for knowledge base question answering, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017, pp. 1321–1331.
- [31] A. P. Parikh, O. Täckström, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: *Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2016, pp. 2249–2255.
- [32] N. Wiratunga, R. Abeyratne, L. Jayawardena, K. Martin, S. Massie, I. Nkisi-Orji, R. Weerasinghe, A. Liret, B. Fleisch, CBR-RAG: case-based reasoning for retrieval augmented generation in LLMs for legal question answering, in: *International Conference on Case-Based*

Reasoning, Springer, 2024, pp. 445–460.

- [33] S. Hakimov, Y. Weiser, D. Schlangen, Evaluating modular dialogue system for form filling using large language models, in: *Proceedings of the 1st Workshop on Simulating Conversational Intelligence in Chat (SCI-CHAT 2024)*, 2024, pp. 36–52.
- [34] B. Ganesan, A. Ravikumar, L. Piplani, R. Bhaumik, D. Padmanaban, S. Narasimhamurthy, C. Adhikary, S. Deshapogu, Automated answer validation using text similarity, *arXiv preprint arXiv:2401.08688* (2024).
- [35] A. Perevalov, A. Gashkov, M. Eltsova, A. Both, Understanding SPARQL Queries: Are we already there? Multilingual Natural Language Generation based on SPARQL Queries and Large Language Models, in: *International Semantic Web Conference*, Springer, 2024, pp. 173–191.
- [36] D. Diefenbach, J. Giménez-García, A. Both, K. Singh, P. Maret, QAnswer KG: designing a portable question answering system over RDF data, in: *European Semantic Web Conference*, Springer, 2020, pp. 429–445. doi:10.1007/978-3-030-49461-2_25.
- [37] K. S. Bisen, S. A. Alemayehu, P. Maret, A. Creighton, R. Gorman, B. Kundi, T. Mgwgi, F. Muhlenbach, S. Dinca-Panaitescu, C. El Morr, Evaluation of Search Methods on Community Documents, Metadata and Semantic Research, 2023, pp. 39–49.
- [38] I. Rybin, V. Korablinov, P. Efimov, P. Braslavski, RuBQ 2.0: An innovated Russian question answering dataset, in: *The Semantic Web: 18th International Conference, ESWC 2021*, Springer, 2021, pp. 532–547.
- [39] L. Siciliani, P. Basile, P. Lops, G. Semeraro, MQALD: Evaluating the impact of modifiers in question answering over knowledge graphs, *Semantic Web 13* (2022).
- [40] D. Zharikova, D. Kornev, F. Ignatov, M. Talimanchuk, D. Evseev, K. Petukhova, V. Smilga, D. Karpov, Y. Shishkina, D. Kosenko, et al., DeepPavlov dream: platform for building generative AI assistants, in: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 2023, pp. 599–607.
- [41] K. Shivashankar, K. Benmaarouf, N. Steinmetz, From graph to graph: AMR to SPARQL, in: *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.
- [42] P. Sen, A. F. Aji, A. Saffari, Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering, in: *Proceedings of the 29th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, 2022, pp. 1604–1619.
- [43] OpenAI, GPT-4 technical report, *arXiv preprint arXiv:2303.08774* (2023). [arXiv:2303.08774](https://arxiv.org/abs/2303.08774).
- [44] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, et al., Qwen2 technical report, *arXiv preprint arXiv:2407.10671* (2024).
- [45] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al., DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, *arXiv preprint arXiv:2501.12948* (2025).
- [46] A. Both, D. Diefenbach, K. Singh, S. Shekarpour, D. Cherix, C. Lange, Qanary – a methodology for vocabulary-driven open question answering systems, in: *The Semantic Web. Latest Advances and New Domains*, Springer International Publishing, 2016, pp. 625–641.
- [47] A. Both, K. Singh, D. Diefenbach, I. Lytra, Rapid engineering of QA systems using the

light-weight Qanary architecture, in: Web Engineering, Springer International Publishing, 2017, pp. 544–548. doi:10.1007/978-3-319-60131-1_40.

- [48] A. Perevalov, A. Both, F. Gudat, P. Bräuning, J. Meesters, L. Gründel, M.-s. Bachmann, S. Z. I. Naser, Qanary Builder: Addressing the reproducibility crisis in question answering over knowledge graphs, in: International Semantic Web Conference (ISWC) – Posters and Demos Track, 2023.