Language Models as SPARQL Query Filtering for Improving the Quality of Multilingual Question Answering over Knowledge Graphs

Aleksandr Perevalov[1(✉)], Aleksandr Gashkov[1], Maria Eltsova[3], and Andreas Both[1,2]

[1] Leipzig University of Applied Sciences, Leipzig, Germany
aleksandr.perevalov@htwk-leipzig.de
[2] DATEV eG, Nuremberg, Germany
[3] CBZ München GmbH, Heilbronn, Germany

Abstract. Question Answering systems working over Knowledge Graphs (KGQA) generate a ranked list of SPARQL query candidates for a given natural-language question. In this paper, we follow our long-term research agenda of providing trustworthy KGQA systems – here – by presenting a query filtering approach that utilizes (large) language models (LMs/LLMs), s.t., correct and incorrect queries can be distinguished. In contrast to the previous work, we address here multilingual questions represented in major languages (English, German, French, Spanish, and Russian), and confirm the generalizability of our approach by also evaluating it on low-resource languages (Ukrainian, Armenian, Lithuanian, Belarusian, and Bashkir). For our experiments, we used the following LMs: BERT, DistilBERT, Mistral, Zephyr, GPT-3.5, and GPT-4. The LMs were applied to the KGQA systems – QAnswer and MemQA – as SPARQL query filters. The approach was evaluated on the multilingual Wikidata-based dataset QALD-9-plus. The experimental results suggest that the KGQA systems achieve quality improvements for all languages when using our query-filtering approach.

Keywords: Question Answering over Knowledge Graphs · Query Validation · Query Candidate Filtering · Trustworthiness

1 Introduction

The main objective of Knowledge Graph Question Answering (KGQA) systems is to provide answers $\mathcal{A}$ that fulfill an informational need of a natural-language (NL) question q, utilizing a Knowledge Graph (KG) [25]. Recent KGQA developments effort in two development paradigms [20,39,41]: (1) the *information extraction paradigm* – aims at retrieving a set of answers directly based on a particular feature space, and (2) the *semantic parsing paradigm* – aims at converting a NL question into a query or a ranked set of *query candidates* that are

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
K. Stefanidis et al. (Eds.): ICWE 2024, LNCS 14629, pp. 3–18, 2024.
https://doi.org/10.1007/978-3-031-62362-2_1

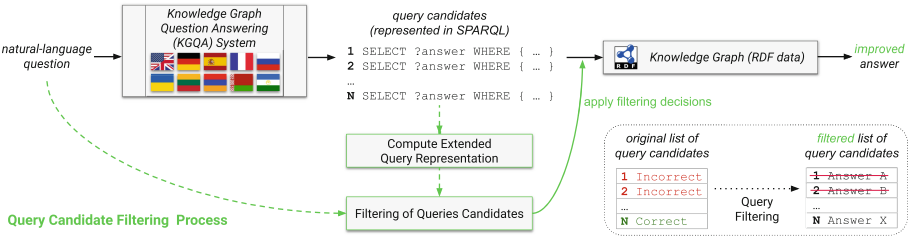

Fig. 1. Big Picture: Query candidate filtering of multilingual KGQA systems.

to be executed on a KG with the sake of retrieving an answer for the question. Let us focus on the semantic parsing paradigm in detail. There, the challenge is that some of the *query candidates might appear incorrect, but still could be prioritized over the correct ones*, leading to a decrease in the quality, and therefore, the trustworthiness of a KGQA system. In this paper, we tackle the previously mentioned challenge by presenting a SPARQL query filtering method that utilizes Language Models (LMs) as filters to differentiate between correct and incorrect SPARQL queries (cf. Fig. 1). In contrast to prior contributions, our work focuses on a diverse set of languages: English, German, Spanish, French, Lithuanian, Russian, Ukrainian, Belarusian, Armenian, and Bashkir. Those languages are provided within the QALD-9-plus [26] – a KGQA benchmark that we use for our experiments. Our main motivation for considering low-resource languages is (1) increasing accessibility of the KGQA systems as only 25.9%[1] of the Web users are English speakers (L1 or L2), and (2) while supporting the low-resource languages, we contribute to the language vitality and the cultural heritage. Therefore, in this work we aim to answer the following *research questions*: RQ1: KGQA system-agnostic – is it possible to establish a quality-improving approach that can be used as an extension to most KGQA systems? RQ2: NL-agnostic – to what degree can the approach be transferred to questions written in different languages (instead of focusing on English input only)?

For our experiments, we use LMs. They are, depending on their type, fine-tuned or instructed as binary classifiers to judge whether a particular SPARQL query can answer a given NL question (*correct*) or cannot (*incorrect*). To ensure high-value insights of the experiments, we use a wide range of LMs. Namely, BERT [15] and DistilBERT [29], open-source instruction-tuned large language models (LLMs) ZEPHYR-7B [35] and Mistral-7B [13], and, finally, proprietary LLMs GPT-3.5 [21] and GPT-4 [22] that represent the current state-of-the-art. In the first experimental stage (S_1), we measure the classification quality on the task of distinguishing between correct and incorrect SPARQL queries provided by the LMs with Precision@1. In the second stage (S_2), we evaluate the effect of applying our query filtering approach on KGQA systems – QAnswer [7] and MemQA (a system introduced in this paper for the sake of providing reference values) – while measuring the QA quality (Precision@1 and Answer

[1] https://www.internetworldstats.com/stats7.htm.

Trustworthiness Score [11]) before and after applying them. Our 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 regarding KGQA, datasets, and language models (Sect. 2). After that, we present in detail our approach in Sect. 3. Section 4 describes the running of our experiments which data are evaluated and analyzed in Sect. 5 followed by the discussion in Sect. 6. Section 7 concludes the paper and outlines future work.

2 Related Work

Multilingual KGQA Systems. A systematic survey of the research field of multilingual KGQA [25] and an overview of the current state of the field for multilingual and cross-lingual subtasks [18] shows that multilingualism in KGQA is still a major challenge, and recently, there is a trend towards that direction. Among well-known multilingual KGQA systems over Wikidata that are currently available, Platypus [23] has support for three languages, DeepPavlov [3] two languages, and QAnswer eight languages [8]. However, only QAnswer provides an extended list of the internal SPARQL query candidates that we can use for our research. Since QAnswer is a real-world system that offers good answer quality (cf. [1,11,28,32]), the semantics of the query candidates are very similar, often almost identical, although the surface form is different (cf. Fig. 6). A share of multilingual solutions is utilizing machine translation (MT) for translating input questions (e.g., [24,34]), which can be easily integrated into a monolingual system, but this way highly depends on the quality of the used MT methods. According to [18] merely translating texts results in a significant drop in performance in some cases and no improvement in others. Other solutions utilize cross-lingual knowledge transfer (e.g., [10,42]), or implementing multilingual LMs (e.g., [9,27]). The authors of [14] proposed an enhanced NL question to SPARQL conversion methodology for a domain ontology-based QA system in Korean and anticipated that, after appropriate modification, this process can be applied to other languages.

Language Models. BERT [6] is designed to learn representations from unlabeled text by joint conditioning on both left and right contexts in all layers. During pretraining, BERT uses masked language modeling (MLM) and next-sentence prediction (NSP). DistilBERT [29] is a general-purpose pre-trained version of BERT distilled on very large batches, leveraging gradient accumulation (up to 4K examples per batch) using dynamic masking and without the NSP objective.

Mistral-7B [13] is a 7-billion-parameter LM that outperforms the previous best 13B model, LLaMA 2 [12], across all tested benchmarks. ZEPHYR-7B [35] is a LM based on Mistral-7B aligned to user intent. The method avoids the use of sampling-based approaches like rejection sampling or proximal preference optimization and distills conversational capabilities with direct preference optimization from a dataset of AI feedback.

The GPT-3 model [2] is a 175 billion parameter autoregressive LLM applied for all tasks without any gradient updates or fine-tuning, with tasks and

few-shot demonstrations specified purely via text interaction with the model. GPT-3 (evolved to GPT-3.5 [40]) showed strong performance on many NLP tasks and benchmarks in the zero-shot, one-shot, and few-shot settings. The GPT-4 model [22] represents a large multimodal model capable of processing image and text inputs and producing text outputs. This is a transformer-based model pre-trained to predict the next token in a document. Both GPT models are multilingual. Despite similar limitations, GPT-4 significantly reduces "hallucinations" relative to the previous GPT-3/GPT-3.5 models [17,19].

KGQA Datasets. Following our research objective, *multilingual KGQA datasets* should meet the following requirements: 1) employing SPARQL over Wikidata as a formal gold-standard query representation; 2) being multilingual (combination of datasets should be multilingual); 3) containing NL representations of questions. However, the recent research [4,5,11,18,25,30,37] indicates the scarcity of the datasets, especially multilingual benchmarks.

To the best of our knowledge, only four existing datasets tackle multiple languages over Wikidata: QALD-9-plus [26], RuBQ 2.0 [28], MCWQ [5], and the recently published Mintaka [31]. However, the latter does not contain a gold standard, i.e., a SPARQL query that would retrieve the correct answer, which is essential for our experiments. MCWQ's languages are Hebrew, Kannada, and Chinese. These languages are rarely employed in research community experiments. The RuBQ 2.0 dataset supports only two languages, MT of questions without any post-editing, and split into very small development (580) and much larger test (2,330) subsets.

The QALD is a well-established benchmark series for multilingual KGQA. QALD-9 [37] contains 558 questions incorporating information of the DBpedia knowledge base. Each question is 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] [26] is an extension of the QALD-9 dataset where extended language support was added, and the translation quality for existing languages was significantly optimized (e.g., for Spanish [33]). The dataset supports English, German, Russian, French, Spanish, Armenian, Belarusian, Lithuanian, Bashkir, and Ukrainian. Additionally, QALD-9-plus added support for the Wikidata knowledge graph. On that account, there is only one dataset matching all our requirements: *QALD-9-plus*.

3 Approach

Our approach revolves around filtering incorrect SPARQL query candidates generated by a KGQA system in response to a NL question. We consider questions in multiple languages, which generalizes our approach more. The core of the approach is to employ fine-tuned or instruction-tuned LMs for binary classification tasks as filters to eliminate incorrect SPARQL queries (see Fig. 1). Let QAS represent a KGQA system, s.t., $QAS^q : NL_q \rightarrow C_q$, where:

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

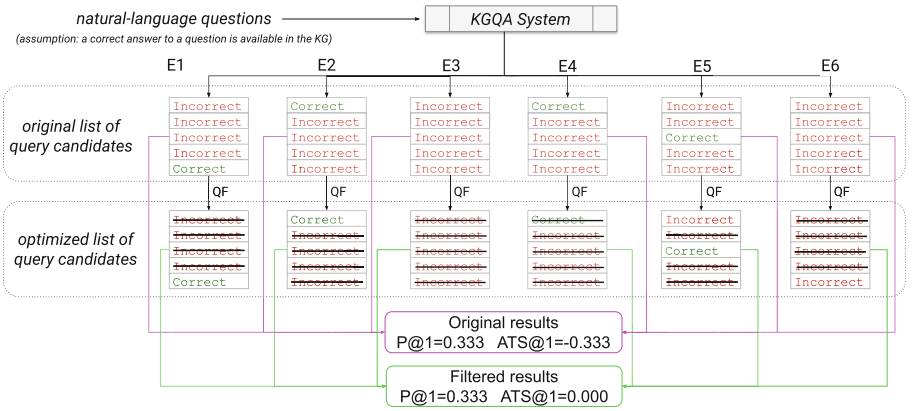

Fig. 2. Impact of Query Filtering on 6 examples (E1 to E6) of lists of 5 candidates evaluated using P@1 and ATS@1. E1 is optimized via the Query Filtering (QF) to a perfect result, in E2 incorrect candidates are removed without changing the result, in E3 and E4 all results are filtered (good for E3 as all incorrect results are eliminated, bad for E4), the optimized results of E5 and E6 have no impact on the quality as an incorrect query candidate is still at the top position. The optimized results have the same P@1 score, however, their trustworthiness is significantly higher.

- Input: NL_q denotes a NL question written in a specific language (e.g., German), where q represents an identifier of the question in a dataset.
- Output: $C_q = \{\text{SPARQL}_1, \text{SPARQL}_2, \ldots, \text{SPARQL}_k\}$ represents the output of the KGQA system for the question q. C_q is an ordered set (i.e., list) of SPARQL query candidates, which may be an empty set, contain one or multiple correct queries, or consist entirely of incorrect queries (6 examples are shown in Fig. 2).

Every question q has a list of *gold standard answers* $\mathcal{A}$ defined by a dataset (can be empty). Following that, a SPARQL query generated 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 produced by a SPARQL_i query $\mathcal{A}^{'}{}_i$ and gold-standard answers $\mathcal{A}_i$ as input, (2) calculates the F1 score over the provided answer sets, and (3) assigns a $label = \{correct\ , incorrect\}$ indicating the correctness of the answer of this query as follows:

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

Therefore, to increase the QA quality by filtering SPARQL query candidates, we need to build a function F that represents a binary classifier, s.t., $F : (NL_i,\ SPARQL_i) \rightarrow label$. Hence, the filtering function F *does not reorder the list but eliminates list items marked as incorrect*. Therefore, 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 utilize LMs (cf. Sect. 2) that are fine-tuned or instruction-tuned as binary classifiers. As many KGs do not provide human-readable URIs of their entities (e.g., Angela Merkel is denoted as `Q567`[3] in Wikidata), we hypothesize that SPARQL queries for such KGs has to be verbalized, i.e., transformed to a NL-like representation while using labels of the corresponding entities from a given KG (e.g., Wikidata).

We distinguish between pre-trained LMs that need to be fine-tuned to a particular downstream task (e.g., BERT) and instruction-tuned LLMs that generate output based on prompts (e.g., Mistral or GPT-3.5). Task-specific LMs require SPARQL queries to be verbalized and used as an input together with a NL question as an input. Instruction-tuned LLMs can utilize the knowledge injection technique in their prompts to draw connections between a URI and its label.

KGQA Systems. We intend to evaluate the efficiency of our approach to real-world KGQA systems. The following selection criteria were defined for the systems: (a) support of multilingual input; (b) answer questions over the Wikidata KG; (c) response with an ordered SPARQL query candidate list.

In addition, to obtain reference values that will fully demonstrate the potential of our approach, we implemented a KGQA system that holds in memory correct SPARQL queries for questions from KGQA benchmarks defined in Sect. 2. Therefore, we call this system *MemQA*[4]. Given a NL question, MemQA returns a list of SPARQL query candidates, where one is the memorized correct query and all the rest are randomly taken from other questions (i.e., incorrect for the given question). The length of this list can be parametrically changed, and the order of produced SPARQL query candidates is random. Hence, all produced SPARQL query candidates are technically sound and defined by humans, as they originate from human-curated benchmarks.

As a correct query candidate is guaranteed, a perfect query validation with a binary classifier would result in a perfect QA quality – this corresponds to a KGQA system capable of providing reference values for our approach.

Evaluation of QA Quality. To measure the effect of the SPARQL query filtering on QA quality, we use the Precision@1 metric, which is calculated before and after applying the approach. We are using the definition of precision recommended by the DICE group that is intended to resolve typical division by zero error in the case of the sum of true positives and true negatives equals 0.0. For this special case, it was defined that if the true positives, false positives, and false negatives are all 0.0, the precision, recall, and F1 score is 1.0 (calculated according to [36]). We calculate P@1 with respect to the mentioned modification. If every candidate is removed, the confusion matrix is filled with all zeroes, and it is impossible to calculate precision because of division by zero. In this case, we suppose P@1 equals zero if any correct candidate was removed in the filtering process and, otherwise, it equals one. As this metric does not

[3] https://www.wikidata.org/wiki/Q567.
[4] https://github.com/WSE-research/memorized-question-answering-system.

take into account unanswerable questions, i.e., $\mathcal{A} = \emptyset$, we also use the Answer Trustworthiness Score ATS (following the definition in [11]) that is specifically designed to reflect the trustworthiness of QA systems, where for all questions q in a dataset D_i a score per question is computed, summed up, and normalized in range of -1 to $+1$:

$$ATS(D_i) = \frac{\sum_{q \in D_i} f(q)}{|D_i|}, \text{ where } f(q) \begin{cases} +1 & \text{if } isCorrect(\mathcal{A}_q, \mathcal{A}'_q) = correct \\ 0 & \text{if } \mathcal{A}'_q = \emptyset \\ -1 & \text{otherwise} \end{cases}$$

ATS Takes into Account Correct, Incorrect, and Empty Answer Sets. 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 average Answer Trustworthiness Score score of 0 can be easily achieved by QA system just by responding with no answer to all questions in D. To have positive Answer Trustworthiness Score, a QA system must give more correct than incorrect answers. Thus, the Answer Trustworthiness Score is more strict than other common metrics and an ideal metrics for measuring the quality of KGQA systems.

4 Experimental Setup

Our experiments are divided into major stages. In the *first stage* (S_1), we conduct binary classification experiments to determine whether a verbalized SPARQL query can answer a given NL question (i.e., correct or incorrect). In the *second stage* (S_2), we apply the binary classifiers to an output of two KGQA systems, MemQA and QAnswer, to validate the produced SPARQL query candidates (i.e., filter out incorrect queries). For both stages, we use the QALD-9-plus (QALD) dataset that has train and test splits.

As described in Sect. 3, we use three groups of LMs, namely: MG_1– BERT-like models, MG_2– open-source instruction-tuned LLMs, and MG_3– commercial instruction-tuned LLMs. In particular, the MG_1 contains the multilingual BERT[5] and multilingual DistilBERT[6], the MG_2 contains Mistral 7B[7] and Zephyr 7B[8], the MG_3 contains GPT-3.5[9] and GPT-4[10]. The detailed experimental setup for S_1 and S_2 is described in the following subsections.

4.1 S_1– Classification

We use micro-averaging-based P@1 for the binary classification task evaluation. The train and test data from QALD-9-plus were prepared as follows. As every

[5] https://huggingface.co/bert-base-multilingual-cased.
[6] https://huggingface.co/distilbert-base-multilingual-cased.
[7] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1.
[8] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta.
[9] https://platform.openai.com/docs/models/gpt-3-5.
[10] https://platform.openai.com/docs/models/gpt-4-and-gpt-4-turbo.

```
SPARQL candidate: SELECT ?uri WHERE { ?uri wdt:P31 wd:Q131436 . }
Input: List all boardgames by GMT.[SEP]{ ?uri instance of board game }
Label: Correct
```

Fig. 3. The example SPARQL query candidate, input tuple, and the corresponding label, which is used to fine-tune and evaluate MG_1 (BERT-like) models. This example is based on the question with "id=1" from the train split of QALD-9-plus (RDF prefixes are omitted).

```
There is a pair of a question and a SPARQL query:
question: List all boardgames by GMT.
query: SELECT ?uri WHERE { ?uri wdt:P31 wd:Q131436 . }
Label for wdt:P31 is instance of.
Label for wd:Q131436 is board game.
Are the question and the query similar? Answer yes or no.
```

Fig. 4. Example of a prompt in English to MG_2 and MG_3 models based on the question with "id=1" from train split of QALD-9-plus.

question with Id q in a dataset has its own gold standard (i.e., correct) SPARQL query, we randomly assigned SPARQL query for other question with Id r ($q \neq r$) from this dataset to form incorrect candidate (cf. *negative sampling*). Hence, for every question, there are two data examples: $[(NL_q,\ SPARQL_q),\ 1]$ – correct or positive example and $[(NL_q,\ SPARQL_r),\ 0]$ – incorrect or negative example. Therefore, the dataset's classes' distribution is balanced.

The models from MG_1 were fine-tuned on data for a binary classification task. Following our approach (see Sect. 3), we verbalized SPARQL queries using Wikidata labels, i.e., they are represented in an NL-like surface form. Hence, the input tuple for the model NL_i and $SPARQL_i^v$ is connected via `[SEP]` token (see Fig. 3). The target *label* values are encoded as a set over $[1, 0]$ respectively. Both models from MG_1 were loaded and trained using the `transformers` [38] library and Hugging Face[11] model hub. While conducting a grid search procedure for epoch tuning, we empirically determined that both BERT and DistilBERT need 4 epochs to achieve optimal quality on our data. Both models were trained with Adam optimizer [16] and batch size equals 16. The hardware setup has the following characteristics: 64 CPUs AMD EPYC 7502P, 96 GB RAM, and no GPU.

The models from MG_2 and MG_3 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 Fig. 4). Based on the aforementioned information, the models from MG_2 and MG_3 are instructed to produce "yes" or "no" corresponding to a correct or incorrect result. The models from MG_2 were loaded and used via the Hugging Face inference endpoint[12] powered by one

[11] https://huggingface.co/models.
[12] https://huggingface.co/inference-endpoints.

```
Input (German): Liste die Brettspiele von GMT auf.
Query candidates:
1: SELECT ?name WHERE { wd:Q23215 wdt:P1477 ?name.}
2: SELECT ?uri WHERE { ?uri wdt:P31 wd:Q131436 . }
```

Fig. 5. MemQA: SPARQL query candidate list with 2 candidates for the German translation of the question in Fig. 4.

```
Input (German): Liste die Brettspiele von GMT auf.
Query candidates:
1: SELECT DISTINCT ?o1 WHERE { wd:Q131436 wdt:P2354> ?o1 . } LIMIT 1000
2: SELECT ?s0  WHERE { VALUES ?s0 { wd:Q12139612> }}
```

Fig. 6. QAnswer: query candidate list with two candidates for the German translation of the question in Fig. 4 (response is simplified, prefixes are omitted)

NVIDIA A10G GPU. The models from MG_3 were used via the official OpenAI Python library[13]. In particular, we utilized the `gpt-3.5-turbo-1106` and `gpt-4` models respectively. The `temperature` parameter was set to 0 and the other parameters were kept with default values.

4.2 S_2– Question Answering

To evaluate the QA quality and the effect of SPARQL query filtering, we calculate such metrics as Precision@1 (P@1) and Answer Trustworthiness Score (ATS@1) before and after the SPARQL query candidate validation. We obtain the query candidates for each question by asking NL questions from the test data splits of QALD-9-plus to the two systems: MemQA (our system for reference values) and QAnswer (real-world system).

We deploy the MemQA system locally and set it up in a way that it produces a different number of query candidates, which is defined before an experiment, namely: 2, 3, 5, 8, 13, 21, 34, 55 (Fibonacci sequence). This is done for obtaining reference values while having diverse sets of SPARQL query candidates in terms of their length. It is worth underlining that MemQA supports every input language, as it is based on the aforementioned QA datasets. The input and output examples of MemQA are shown in Fig. 5 and 6.

The QAnswer system produces different numbers of query candidates for questions: from 0 to 60 as observed empirically. The system also does not cover all the languages presented in the test datasets, as described in Sect. 2. Therefore, we used four languages both presented in the dataset and supported by QAnswer (English, German, Russian, and Spanish) which have enough data for model training. The QAnswer system was used via its public API[14].

[13] https://github.com/openai/openai-python.
[14] https://backend.app.qanswer.ai/swagger-ui/index.html.

5 Evaluation and Analysis

5.1 S_1– Classification

When working with multilinguality, we determine the best-performing model while aiming at two objectives: the average F1 score and the standard deviation (stdev) of a particular model over all languages [25]. Hence, the joint objective is to achieve the highest average F1 score and lowest standard deviation of the F1 score values. While analyzing the results between the different model groups, the MG_2 has significantly worse quality than MG_1 and MG_3. In turn, MG_1 and MG_3 have comparable quality, however, GPT-4 achieves equally high F1 score on most of the languages despite Bashkir. The BERT, DistilBERT, and GPT-3.5 models achieve the highest quality on high-resource languages (English, German, Russian, French, and Spanish) while the quality of the rest languages has drawdowns. Therefore, BERT-like models and closed-source GPT models significantly outperform open-source LLMs on our binary classification task setting (see online appendix[15]).

5.2 S_2– Question Answering

In this subsection, we present the evaluation results of the two QA systems, MemQA and QAnswer[16]. While applying the SPARQL query filtering approach, we analyze its effect on QA quality. As *MemQA* simulates the behavior of an almost "ideal" KGQA system by having at least one correct SPARQL query in every list of query candidates, we use its results as reference values to show what impact is achievable with SPARQL query filtering for QA in ideal conditions.

In Table 1 we present the results for Answer Trustworthiness Score @1 and Precision@1 calculated when applying our approach on MemQA.

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 KGQA system MemQA. Just questions in the Bashkir language do not fully benefit from the filtering process. The Precision@1 results in Table 1 indicate a significant improvement in MG_1 and GPT 4 models excluding Armenian for BERT and French for DistilBERT. GPT 3.5 model improves quality for major languages but decreases it for low-resources languages (Belorussian, Bashkir, and Armenian). As these are average values from the experiments, we can conclude that the approach works in general.

The Precision@1 results in Fig. 7 show that the MG_1 and MG_3 models demonstrate an improvement of Precision@1 after applying the filtering approach in half of the experimental cases on the QALD-9-plus dataset.

[15] https://anonymous.4open.science/r/QAfiltering-3C5E/data/resources/classification-pareto.pdf.

[16] The raw data is available in the online appendix at: https://anonymous.4open.science/r/QAfiltering-3C5E/.

Table 1. Results of our filtering method on the MemQA system

Language	No filtering	BERT	DistilBERT	Mistral	Zephyr	GPT 3.5	GPT 4
Answer Trustworthiness Score @ 1							
en	–0.580	0.719	0.720	0.362	–0.264	0.318	**0.904**
de	–0.603	0.524	0.605	–0.640	–0.241	0.330	**0.862**
es	–0.608	0.791	0.736	–0.079	–0.110	0.073	**0.853**
ru	–0.555	0.337	0.338	–0.535	0.018	0.092	**0.783**
fr	–0.574	**0.800**	–0.200	0.113	–0.296	0.427	0.760
be	–0.615	0.137	0.343	–0.025	–0.032	–0.209	**0.883**
uk	–0.654	0.248	0.398	0.156	–0.005	–0.276	**0.922**
ba	–0.597	–0.453	–0.056	–0.015	–0.097	–0.555	**0.000**
lt	–0.579	0.491	0.346	–0.298	–0.181	–0.178	**0.882**
hy	–0.496	0.053	0.305	–0.021	0.095	–0.226	**0.832**
Precision @ 1							
en	0.210	0.854	0.830	0.415	0.209	0.365	**0.904**
de	0.198	0.751	0.747	0.167	0.174	0.520	**0.862**
es	0.196	**0.880**	0.819	0.029	0.006	0.247	0.853
ru	0.223	**0.895**	0.878	0.225	0.458	0.706	**0.895**
fr	0.213	**0.827**	0.393	0.141	0.191	0.453	0.760
be	0.193	0.548	0.559	0.000	0.122	0.106	**0.901**
uk	0.173	0.615	0.648	0.458	0.080	0.226	**0.923**
ba	0.201	0.271	**0.294**	0.002	0.064	0.078	0.000
lt	0.211	0.634	0.542	0.043	0.197	0.410	**0.884**
hy	0.252	0.053	0.505	0.263	0.147	0.137	**0.863**

There are two reasons for the outlier results for Russian, even before filtering, the P@1 is very high. Firstly, QAnswer produces up to 60 candidates for other languages while generating only 3 candidates for Russian in most cases. Secondly, the task of distinguishing correct/incorrect candidates usually becomes trivial for Russian, e.g., for the question "Какой часовой пояс в Солт-Лейк-Сити?" (What is the time zone of Salt Lake City?) the candidates are:

- `SELECT DISTINCT ?o1 WHERE wd:Q23337 wdt:P421 ?o1 . LIMIT 1000`
- `SELECT DISTINCT ?o1 WHERE ?s1 wdt:P31 ?o1 . LIMIT 1000`
- `SELECT DISTINCT ?s1 ?o1 WHERE ?s1 wdt:P31 ?o1 . LIMIT 1000`

The first one is correct, while the two others are just nonsense (the label for `wdt:P31` is "instance of").

In Table 2 we present the average results for Answer Trustworthiness Score achieved on *QAnswer*. The numbers do not appear to be conclusive, but this is because the quality of GPT-4 decreases when the number of query candidates exceeds 6 (see the online appendix. Figure 7 demonstrates the Precision@1 values for datasets and models before and after applying SPARQL query filtering.

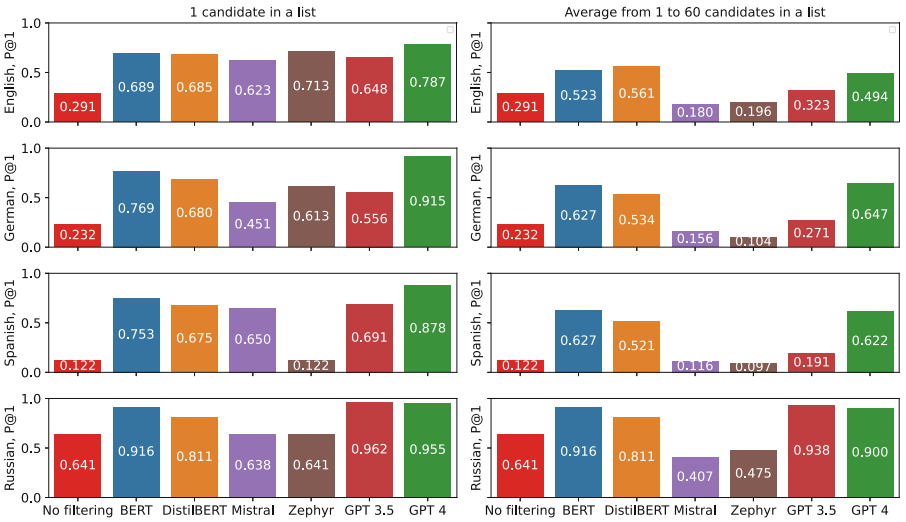

Fig. 7. Precision@1 values for the QAnswer system. The left-hand side of the figure demonstrates the results when the lists of query candidates are cut off to 1 (i.e., there is no second candidate that might move to the top of the list), while the right-hand side shows average values for the candidates' lists from 1 to 60. Each bar demonstrates the value of a particular model. The "No filtering" column shows the metric value without our approach.

Table 2. Results of our filtering method on the QAnswer system aggregated over all different lengths of query candidate lists (from 1 to 60) (see Sect. 4.2).

Language	No filtering	BERT	DistilBERT	Mistral	Zephyr	GPT 3.5	GPT 4
Answer Trustworthiness Score @ 1							
en	–0.418	–0.160	**–0.119**	–0.648	–0.627	–0.375	–0.169
de	–0.535	–0.123	–0.223	–0.694	–0.814	–0.502	**–0.080**
es	–0.756	**–0.206**	–0.339	–0.791	–0.861	–0.684	–0.208
ru	0.282	0.592	0.487	–0.176	–0.057	**0.613**	0.571
Precision @ 1							
en	0.291	0.523	**0.561**	0.180	0.196	0.323	0.494
de	0.232	0.627	0.534	0.156	0.104	0.271	**0.647**
es	0.122	**0.627**	0.521	0.116	0.097	0.191	0.622
ru	0.641	**0.916**	0.811	0.407	0.475	0.938	0.900

6 Discussion

Our results show a strong impact of our approach to the questions in all languages. However, there are some improvement's outliers wrt. languages that are rarely used: Belarusian, Lithuanian, Armenian, and Bashkir as well as to some degree Ukrainian. A post-experiment analysis showed that many questions could

not be processed in our approach as labels for the resources were not available, leading to an automatic acceptance of the question (i.e., the filtering method was not applied). This observation highlights a crucial problem while aiming for the accessibility of information from the Web of Data for all humans (cf. [24]). Hence, we can derive here the need for completing the Linked Open Data Cloud at least concerning the resource labels, s.t., a wider information accessibility is supported. Given the poor quality of MG_2 one might argue that the used prompt – although using a straight-forward text – has caused the problems regarding these models. Given additional manual experiments, we tentatively assume that such an LLM-specific prompt optimization would not significantly change the result. A similar point could be made if only English prompts were used. A language-specific prompt might lead to a quality improvement. However, these topics might need additional evaluation beyond the scope of this paper.

7 Conclusions and Future Work

In this paper, we presented an approach for improving the quality of Question Answering over Knowledge graphs. In contrast to other research, we did not present a new KGQA algorithm but a general approach on how to improve the answer quality. In particular, our approach is capable of removing incorrect query candidates, s.t., the number of incorrect results shown to the users is significantly reduced – a fact that strongly increases the trustworthiness of such systems. Additionally, we dedicated our work to developing an approach that also applies to non-English questions. In particular, we evaluated rarely used languages to address the need of people to access information from KGQA systems using their native language (which is not English for most of the worldwide population) without using machine translation. The unique features of our approach are:

(1) The system-agnostic process built on top of the query candidates represented uses the SPARQL format as it is typical in the field of KGQA. Hence, our approach can be applied to existing systems to improve their answer quality (i.e., their trustworthiness). Our experiment provides a rough range of possible improvements to KGQA systems by our approach.
(2) We followed a language-agnostic approach. Hence, it can be transferred to other languages without changing the process. The only requirement is the representation of language-specific labels for the relevant labels in the considered Knowledge Graph. Our results show that our approach can be applied to other languages and will improve the quality of questions represented in other languages as well, with a similar increase of trustworthiness as for English.
(3) Both LLMs and smaller language models can be used for our approach. So that users have the choice of which technology they use. Our experiments show a strong quality improvement for two out of three LM categories used for our experiments. We observed an advantage of closed-source LLMs (which are presumably an order of magnitude larger than the used open-source LLMs), however, they might not apply to all use cases (e.g., because

of privacy issues or as they might imply a significantly higher investment of computing time or cost-per-interaction).

Future work may require experiments with a language-specific prompt, as well as an LLM-specific prompt. Our approach could be extended by using additional KG properties. Additionally, a promising direction for improving the results would be to solve the problem of labels' non-availability of the resources.

References

1. Bisen, K.S., et al.: Evaluation of search methods on community documents. In: Garoufallou, E., Vlachidis, A. (eds.) MTSR 2022, vol. 1789, pp. 39–49. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-39141-5_4
2. Brown, T., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
3. Burtsev, M., et al.: DeepPavlov: open-source library for dialogue systems. In: Proceedings of ACL 2018, System Demonstrations, pp. 122–127. ACL (2018)
4. Cui, R., Aralikatte, R., Lent, H., Hershcovich, D.: Multilingual compositional Wikidata questions. arXiv preprint arXiv:2108.03509 (2021)
5. Cui, R., Aralikatte, R., Lent, H., Hershcovich, D.: Compositional generalization in multilingual semantic parsing over Wikidata. Trans. ACL **10** (2022)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
7. Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the semantic web. Semant. Web **11**, 421–439 (2020)
8. Diefenbach, D., Giménez-García, J., Both, A., Singh, K., Maret, P.: QAnswer KG: designing a portable question answering system over RDF data. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 429–445. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_25
9. Efimov, P., Boytsov, L., Arslanova, E., Braslavski, P.: The impact of cross-lingual adjustment of contextual word representations on zero-shot transfer. In: Kamps, J., et al. (eds.) ECIR 2023, vol. 13982, pp. 51–67. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-28241-6_4
10. Evseev, D.: Query generation for answering complex questions in Russian using a syntax parser. Sci. Techn. Inf. Process. **49**(5) (2022)
11. Gashkov, A., Perevalov, A., Eltsova, M., Both, A.: Improving question answering quality through language feature-based SPARQL query candidate validation. In: Groth, P., et al. (eds.) ESWC 2022, vol. 13261, pp. 217–235. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-06981-9_13
12. Jayaseelan, N.: LLaMA 2: the new open source language model (2023). https://www.e2enetworks.com/blog/llama-2-the-new-open-source-language-model
13. Jiang, A.Q., et al.: Mistral 7B. arXiv preprint arXiv:2310.06825 (2023)
14. Jung, H., Kim, W.: Automated conversion from natural language query to SPARQL query. J. Intell. Inf. Syst. **55**(3), 501–520 (2020)
15. Kenton, J.D.M.W.C., Toutanova, L.K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, vol. 1 (2019)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Koubaa, A.: GPT-4 vs. GPT-3.5: a concise showdown. Preprints (2023)

18. Loginova, E., Varanasi, S., Neumann, G.: Towards end-to-end multilingual question answering. Inf. Syst. Front. **23**, 227–241 (2021)
19. McIntosh, T.R., Liu, T., Susnjak, T., Watters, P., Ng, A., Halgamuge, M.N.: A culturally sensitive test to evaluate nuanced GPT hallucination. IEEE Trans. Artif. Intell. **1**(01), 1–13 (2023)
20. McKenna, N., Sen, P.: KGQA without retraining. In: ACL 2023 Workshop on SustaiNLP (2023)
21. OpenAI: Introducing ChatGPT (2022). https://openai.com/blog/chatGPT
22. OpenAI: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
23. Pellissier Tanon, T., de Assunção, M.D., Caron, E., Suchanek, F.M.: Demoing platypus – a multilingual question answering platform for Wikidata. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 11155, pp. 111–116. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98192-5_21
24. Perevalov, A., Both, A., Diefenbach, D., Ngonga Ngomo, A.C.: Can machine translation be a reasonable alternative for multilingual question answering systems over knowledge graphs? In: ACM Web Conference 2022, WWW 2022. ACM (2022)
25. Perevalov, A., Both, A., Ngomo, A.C.N.: Multilingual question answering systems for knowledge graphs-a survey. Semant. Web J. (2023)
26. Perevalov, A., Diefenbach, D., Usbeck, R., Both, A.: QALD-9-plus: a multilingual dataset for question answering over DBpedia and Wikidata translated by native speakers. In: International Conference on Semantic Computing (ICSC) (2022)
27. Razzhigaev, A., Salnikov, M., Malykh, V., Braslavski, P., Panchenko, A.: A system for answering simple questions in multiple languages, pp. 524–537. ACL (2023)
28. Rybin, I., Korablinov, V., Efimov, P., Braslavski, P.: RuBQ 2.0: an innovated Russian question answering dataset. In: Verborgh, R., et al. (eds.) ESWC 2021. LNCS, vol. 12731, pp. 532–547. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77385-4_32
29. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
30. Saxena, A., Chakrabarti, S., Talukdar, P.: Question answering over temporal knowledge graphs. arXiv preprint arXiv:2106.01515 (2021)
31. Sen, P., Aji, A.F., Saffari, A.: Mintaka: a complex, natural, and multilingual dataset for end-to-end question answering. In: 29th International Conference on Computational Linguistics, pp. 1604–1619 (2022)
32. Siciliani, L., Basile, P., Lops, P., Semeraro, G.: MQALD: evaluating the impact of modifiers in question answering over knowledge graphs. Semant. Web **13**(2) (2022)
33. Soruco, J., Collarana, D., Both, A., Usbeck, R.: QALD-9-ES: a Spanish dataset for question answering systems. In: Studies on the Semantic Web, pp. 38–52. IOS Press BV (2023)
34. Srivastava, N., et al.: Lingua franca - entity-aware machine translation approach for question answering over knowledge graphs. In: Knowledge Capture Conference. ACM (2023)
35. Tunstall, L., et al.: Zephyr: direct distillation of LM alignment. arXiv preprint arXiv:2310.16944 (2023)
36. Usbeck, R., et al.: Gerbil: general entity annotator benchmarking framework. In: 24th International Conference on World Wide Web, WWW 2015. (2015)
37. Usbeck, R., Gusmita, R.H., Ngomo, A.C.N., Saleem, M.: 9th challenge on question answering over linked data (QALD-9). In: Semdeep/NLIWoD@ISWC (2018)
38. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Empirical Methods in NLP: System Demonstrations, pp. 38–45. ACL (2020)

39. Xu, S., Culhane, T., Wu, M.H., Semnani, S.J., Lam, M.S.: Complementing GPT-3 with few-shot sequence-to-sequence semantic parsing over Wikidata. arXiv preprint arXiv:2305.14202 (2023)
40. Ye, J., et al.: A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. arXiv preprint arXiv:2303.10420 (2023)
41. Zhang, C., Lai, Y., Feng, Y., Zhao, D.: A review of deep learning in question answering over knowledge bases. AI Open **2**, 205–215 (2021)
42. Zhou, Y., Geng, X., Shen, T., Zhang, W., Jiang, D.: Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In: NAACL: Human Language Technologies, pp. 5822–5834. ACL (2021)