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

Aleksandr Perevalov[1,*], Aleksandr Gashkov[1], Maria Eltsova[3] and Andreas Both[1,2]

[1]*Leipzig University of Applied Sciences, Leipzig, Germany*
[2]*DATEV eG, Nuremberg, Germany*
[3]*CBZ München GmbH, Heilbronn, Germany*
E-mail: aleksandr.perevalov@htwk-leipzig.de; aleksandr.gashkov@htwk-leipzig.de; maria.eltsova@gmail.com; andreas.both@datev.de; andreas.both@htwk-leipzig.de
[*]*Corresponding Author*

Received 02 December 2024; Accepted 26 March 2025

Abstract

Question answering is an approach to retrieving information from a knowledge base using natural language. Within question answering systems that work over knowledge graphs (KGQA), a ranked list of SPARQL query candidates is typically computed for the given natural-language input, where the top-ranked query should reflect the intention and semantics of the given user's question. This article follows our long-term research agenda of providing trustworthy KGQA systems by presenting an approach for filtering incorrect queries. Here, we employ (large) language models (LMs/LLMs) to distinguish between correct and incorrect queries. The main difference to the previous work is that we address here multilingual questions represented in major languages (English, German, French, Spanish, and Russian), and confirm the generalizability of the approach by also evaluating it on some

Journal of Web Engineering, Vol. 24_4, 563–592.
doi: 10.13052/jwe1540-9589.2444

low-resource languages (Ukrainian, Armenian, Lithuanian, Belarusian, and Bashkir). The considered LMs (BERT, DistilBERT, Mistral, Zephyr, GPT-3.5, and GPT-4) were applied to the KGQA systems – QAnswer (real-world system) and MemQA (idealized system) – as SPARQL query filters. The approach was evaluated on the multilingual dataset QALD-9-plus, which is based on the Wikidata knowledge graph. The experimental results imply that the considered 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, question answering quality, 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 question q, utilizing a knowledge graph (KG) [37]. Recent KGQA developments effort has focused on two development paradigms [24, 56, 58]: (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 natural-language question into a query or (more generalized) a ranked set of *query candidates* that are to be executed on a KG with the aim of retrieving an answer for the given 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, impacting the trustworthiness of a KGQA system.

We tackle the previously mentioned challenge by presenting a filtering method for SPARQL queries that utilizes language models (LMs) as filters to

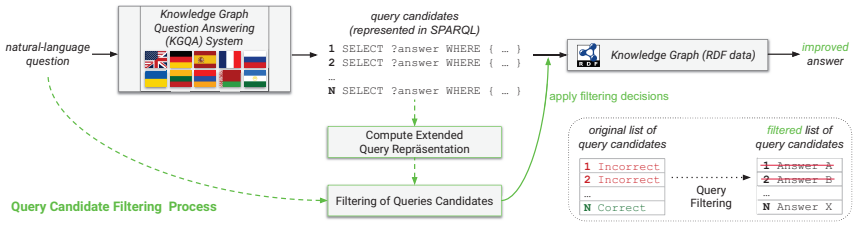

Figure 1 Big picture: Query candidate filtering embedded in multilingual knowledge graph question answering systems.

differentiate between correct and incorrect SPARQL queries (cf. Figure 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 [38] – a well-known 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. Despite people generally preferring to use their native language while searching [47], some groups of users have to search in English (e.g., academic users [29]). Due to this observation, we derive the need for particular support for the KGQA systems dedicated to answering questions represented in rarely used languages (i.e., questions represented in non-English languages).

Here, we also focus on the trustworthiness of QA systems, which obviously depends on the quality of the answers with a specific focus on KGQA, which is dedicated to the task of generating for a given natural-language question a SPARQL query that can be used to retrieve the correct answer from the intended knowledge graph. However, often forgotten is the fact that trustworthiness is particularly important if the KGQA quality is lower, as is typically the case for non-English KGQA systems. Therefore, in this work, we aim to answer the following *research questions*:

RQ1 *Agnostic regarding KGQA systems* – is it possible to establish a quality-improving approach that can be used as an extension to most KGQA systems?

RQ2 *Agnostic regarding the natural language* – 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 language models. They are, depending on their type, fine-tuned or instructed as binary classifiers to judge whether a particular SPARQL query can answer a given natural-language question (*correct*) or cannot (*incorrect*). To ensure high-value insights of the experiments, we use a wide range of language models: Namely, BERT [8] and DistilBERT [44], open-source instruction-tuned large language models (LLMs) ZEPHYR-7B [51] and Mistral-7B [16], and, finally, proprietary LLMs GPT-3.5 [30] and GPT-4 [31] that represent the current state-of-the-art. In the first

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

experimental stage (S_1), we measure the classification quality on the task of distinguishing between correct and incorrect SPARQL queries provided by the language models with Precision@1. In the second stage (S_2), we evaluate the effect of applying our query filtering approach on KGQA systems – QAnswer [9] and MemQA (a system introduced in this articlefor the sake of providing reference values) – while measuring the QA quality (Precision@1 and Answer Trustworthiness Score [13]) before and after applying them. Our results show a strong impact on the quality of both scores and all languages.

This article has the following structure. First, we describe the related work (Section 2). After that, we present in detail our approach in Section 3. Section 4 describes the data preparation process and the running of our experiments, whose data are evaluated and analyzed in Section 5 followed by the discussion in Section 6. Section 7 concludes the article and outlines future work.

2 Related Work

2.1 Multilingual KGQA Approaches and Systems

A systematic survey of the research field of multilingual KGQA [37] and an overview of the current state of the field for multilingual and cross-lingual subtasks [22] shows that multilingualism in KGQA is still a major challenge (cf. [40]), and recently, there is a trend towards that direction, especially in the field of KGQA. Another problem being pointed out in the research community is the saturation of the KGQA field with work on English data, due to both the inherent challenges of translating datasets and the reliance on English-only knowledge bases [6].

The detailed review of the multilingual KGQA systems [37] indicates that 11 out of 17 systems' results are not reproducible due to the outdated demo APIs and source code or their absence. Among well-known multilingual KGQA systems over the Wikidata knowledge graph that are currently available, Platypus [32] has support for three languages, DeepPavlov [5] two languages, and QAnswer eight languages [10]. However, only QAnswer provides an extended list of the internal SPARQL query candidates that we can use for our research. Since the QAnswer approach is available as a real-world system that offers good answer quality (cf. [1,13,43,48]), the semantics of the query candidates are often very similar, many times almost identical, although the surface form is different (cf. Figure 6). Recently designed systems and methods like *Tiresias* [25] and a research prototype supporting bilingual QA

over DBpedia abstracts in Greek or English language, or *M3M* [42] have some limitations, and need additional evaluation or extension of the technical accessibility.

A share of multilingual solutions is utilizing machine translation (MT) for translating input questions (e.g., [35, 50]), which can be easily integrated into a monolingual system. However, this way highly depends on the quality of the used MT methods: According to [22], merely translating texts results in a significant drop in performance in some cases and no improvement in others. Moreover, the linguistic studies in indirect translation prove that despite some positive results in the literary translation, it entails a loss of detail [20], intercultural text transfer is often mediated by dominant systems [14] and some other flaws resulting in a recommendation by UNESCO (1976) suggesting that indirect translation should be used "only where absolutely necessary" [52].

Other solutions utilize cross-lingual knowledge transfer (e.g., [59]), or implementing multilingual LMs (e.g., [11, 41, 42]). The authors of [17] proposed an enhanced natural-language question to SPARQL conversion methodology for a QA system in Korean based on a domain ontology and anticipated that, after appropriate modification, this process can be applied to other languages.

2.2 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 natural-language representations of questions. However, the recent research [6, 7, 13, 22, 37, 45, 53] indicates the scarcity of datasets, especially multilingual benchmarks.

To the best of our knowledge, only four existing datasets tackle multiple languages over Wikidata: QALD-9-plus [38], RuBQ 2.0 [43], MCWQ [7], and Mintaka [46]. 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 English, Hebrew, Kannada, and Chinese. Three of its languages are rarely (or not) employed in research community experiments and aren't supported by most QA systems. The RuBQ 2.0 dataset supports only two languages, machine-translated questions without any post-editing, split into a small development set (580 questions) and a much larger test set (2330 questions).

QALD is a well-established benchmark series for multilingual KGQA. QALD-9 [53] contains 558 questions incorporating information of the DBpedia knowledge graph. 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] [38] is an extension of the QALD-9 dataset where extended language support was added, and the translation quality for existing languages was significantly optimized by validations of native speakers. The dataset supports English, German, Russian, French, Spanish, Armenian, Belarusian, Lithuanian, Bashkir, and Ukrainian (Spanish was added via [49]). 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*.

2.3 Verbalization of SPARQL Queries

The previous work on the topic of conversion of SPARQL queries to natural language was mostly based on grammar rules and relatively small language models (cf. [21, 26–28] etc.). A new approach to enhancing answers verbalization using LLMs was recently introduced by the DICE group[3]. The developers found that fine-tuning language models and introducing additional knowledge, such as SPARQL queries, to achieve state-of-the-art results in verbalizing answers from KGQA systems. Therefore, their approach can be used to generate answers' verbalization for different KGQA systems, including dialogue systems or voice assistants.

However, the research in the field of SPARQL query verbalization is mostly aimed at generating an answer from the given SPARQL query (there are only a few exceptions, e.g., [39]). Our idea, in contrast, is to use verbalization to validate if a SPARQL query generated by a QA system corresponds to the question. We define verbalization as a process of generating natural-language text based on the SPARQL queries or as a natural-language form of a SPARQL query. Using our approach, it is possible to create natural-language representations for public or private KGs while providing the labels of resources. This approach does not generate a lot of possible answers (mostly incorrect) while filtering out the incorrect query candidates just

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

[3] Available online at https://papers.dice-research.org/2024/SEMANTICS_Answers_Verbalization/public.pdf

by their natural-language representation. In our previous paper [12], we described three different approaches of verbalization[4]:

- Approach x_1 provided using well-formed natural language written by a human as a baseline (e.g., *John Kennedy was assassinated*).
- Approach x_2 computed using Natural Language Generation (NLG) considering the contained facts (e.g., *The John F. Kennedy's death cause is Assassination of John F. Kennedy*).
- Approach x_3 computed using a bag-of-labels approach of available entities (e.g., *John F. Kennedy death cause Assassination of John F. Kennedy*).

Moreover, we found out that our method demonstrated its effectiveness by eliminating incorrect answer candidates based only on natural-language representations of questions and possible answers, even the results of x_3 provided strong evidence that the QA quality was significantly increased while using it in comparison to unvalidated answer candidate sets. Therefore, for our further experiments, we use the easiest for realization "bag-of-labels approach of available entities".

3 Approach

Our approach revolves around filtering incorrect SPARQL query candidates generated by a KGQA system in response to a natural-language 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 Figure 1). 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: $C_q = \{\text{SPARQL}_1, \text{SPARQL}_2, \ldots, \text{SPARQL}_k\}$ represents the output of the KGQA system for the question identifier q. C_q is an ordered set (i.e., list without duplicates) of SPARQL query candidates, which may be an empty set, contain one or multiple correct queries, or

[4]The examples of different verbalization approaches are given for the SPARQL query `dbr:John_F._Kennedy`, `dbo:deathCause`, and `dbr:Assassination_of_John_F._Kennedy`

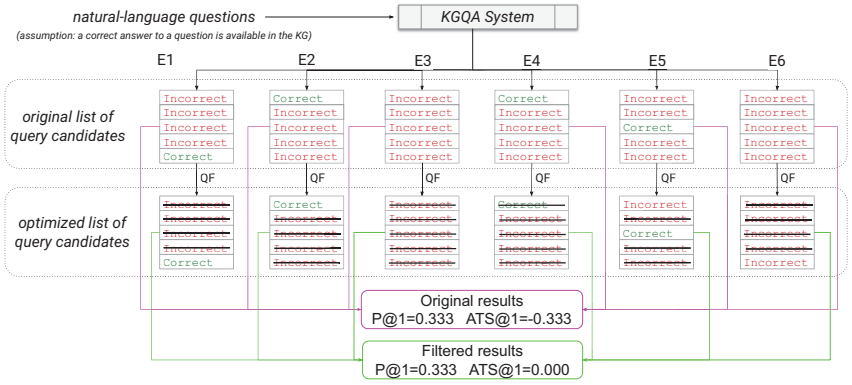

Figure 2 Impact of query filtering on six examples (E1 to E6) of lists of five candidates evaluated using P@1 and ATS@1. E1 is optimized via query filtering (QF) to a perfect result, in E2 incorrect candidates are removed without changing the result, in E3 and E4 all results are filtered (good for E3 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.

consist entirely of incorrect queries (six of such scenarios are shown in Figure 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., $F : (\text{NL}_i,\ \text{SPARQL}_i) \rightarrow label$. 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 with a better position are removed.

Language models. **BERT** [8], which stands for bidirectional encoder representations from transformers, is designed to learn representations from an 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). BERT achieved state-of-the-art performance on a large suite of sentence-level and token-level tasks, outperforming many task-specific architectures. A distinctive feature of BERT is its unified architecture across different tasks. The pre-trained BERT model is utilized for the downstream tasks (e.g., sequence classification) by connecting additional output layers.

DistilBERT [44] 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. DistilBERT is 40% smaller, 60% faster that retains 97% of the language understanding capabilities.

Mistral-7B [16] is a 7-billion-parameter LM. It demonstrates that a carefully designed language model is able, firstly, to deliver high performance while maintaining an efficient inference and, secondly, to compress knowledge more than what was previously thought. Mistral-7B outperforms the previous best 13B model, LLaMA 2 [15], across all tested benchmarks.

ZEPHYR-7B [51] is an LM based on Mistral-7B aligned to user intent. The developers consider the problem of alignment distillation from an LLM onto a smaller pre-trained model. 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 model should demonstrate that LMs may compress knowledge more than what was previously thought. The resulting model ZEPHYR-7B sets a new state-of-the-art for 7B parameter chat models and even outperforms LLAMA2-CHAT-70B on MT-Bench.

The **GPT-3** model [4] 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 [57]) showed strong performance on many NLP tasks and benchmarks in the zero-shot, one-shot, and few-shot settings, in some cases nearly matching the performance of state-of-the-art fine-tuned systems, as well as generating high-quality samples and strong qualitative performance at tasks defined on-the-fly. There are some limitations and flaws of the model that do not affect our method: on text synthesis, repeating themselves semantically at the document level, starting to lose coherence over

sufficiently long passages, contradicting itself, and occasionally containing non-sequitur sentences or paragraphs; lacking a large amount of context about the world; challenging to deploy due to its size.

The **GPT-4** model released in 2023 [31] 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, with a wide range of languages covering different geographic regions and scripts. Despite similar limitations (e.g., it is not fully reliable, has a limited context window, and does not learn from experience), GPT-4 significantly reduces "hallucinations" relative to the previous GPT-3/GPT-3.5 models [19, 23]. According to the developers' experimental results, the GPT-4 scores 19 percentage points higher than the latest GPT-3.5 on internal, adversarial-designed factuality evaluations.

We distinguish between pre-trained language models 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).

Verbalization and binary classification of SPARQL queries. To create the filtering function F, we utilize language models 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`[5] in Wikidata), we hypothesize that SPARQL queries for such KGs have to be verbalized, i.e., transformed to a representation that is similar to natural language while using labels of the corresponding entities from a given KG (e.g., Wikidata).

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. The only real-world KGQA systems matching all criteria is QAnswer (see Section 2 and [10]).

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. Therefore, we call this system *MemQA*[6]. Given a natural-language question, MemQA returns a list of SPARQL query candidates, where one is

[5] https://www.wikidata.org/wiki/Q567

[6] https://github.com/WSE-research/memorized-question-answering-system

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, which 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 are 1.0 (calculated according to [54]). 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 take into account unanswerable questions, i.e., $\mathcal{A} = \emptyset$, we also use the answer trustworthiness score, ATS (following the definition in [13]) which 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 of 0 can be easily achieved by a QA system just by responding with no answer to all questions in D. To achieve a positive answer trustworthiness score, a QA system must give more correct than incorrect answers. Thus, the ATS is more strict than other common metrics and an ideal metric 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 natural-language question (i.e., it returns a correct or incorrect result). 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 dataset that provides the required train and test splits.

As described in Section 3, we use three groups of LMs, namely: model group MG_1 – BERT-like models, MG_2 – open-source instruction-tuned LLMs, and MG_3 – commercial instruction-tuned LLMs. In particular, MG_1 contains the multilingual BERT[7] and multilingual DistilBERT[8],MG_2 contains Mistral 7B[9] and Zephyr 7B[10], MG_3 contains GPT-3.5[11] and GPT-4[12]. 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 question with the identifier q in a dataset has its own gold standard (i.e., correct) SPARQL query, we randomly assigned the SPARQL query for other questions with the identifier r ($q \neq r$) from this dataset to form an incorrect candidate (cf. *negative sampling*). Hence, for every question, there are two data examples: $[(\text{NL}_q,\ \text{SPARQL}_q),\ 1]$ – correct or positive examples and $[(\text{NL}_q,\ \text{SPARQL}_r),\ 0]$ – 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 Section 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 Figure 3). The target $label$ values are encoded as a set over

[7] https://huggingface.co/bert-base-multilingual-cased
[8] https://huggingface.co/distilbert-base-multilingual-cased
[9] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
[10] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta
[11] https://platform.openai.com/docs/models/gpt-3-5
[12] 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
```

Figure 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.
```

Figure 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.

$[1, 0]$ respectively. Both models from MG_1 were loaded and trained using the `transformers` [55] library and Hugging Face[13] model hub. While conducting a grid search procedure for epoch tuning, we empirically determined that both BERT and DistilBERT need four epochs to achieve optimal quality on our data. Both models were trained with the Adam optimizer [18] and the 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 Figure 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[14] powered by one NVIDIA A10G GPU. The models from MG_3 were used via the official OpenAI Python library[15]. In particular, we utilized the `gpt-3.5-turbo-1106` and `gpt-4` models respectively. The

[13] https://huggingface.co/models
[14] https://huggingface.co/inference-endpoints
[15] https://github.com/openai/openai-python

```
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 . }
```

Figure 5 MemQA: SPARQL query candidate list with two candidates for the German translation of the question in Figure 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> }}
```

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

`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 natural-language 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 Figures 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 Section 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.[16]

4.3 Data Preparation Process

In a pre-study, we observed that the benchmark data contains noise and cannot be used as it is.

The dataset cleanup process concerning only errors in the benchmarks consists of two essential steps: dataset preparation and candidates' preparation. Both steps implement a similar procedure. The dataset preparation process removes all records with errors from the dataset. Removed records are not considered in all further processing. The candidates' preparation process is the same. First, we need to evaluate the correctness of each candidate by comparing the results of the gold standard query and the candidate query on Wikidata. If either of the queries cannot be executed, we cannot evaluate the candidate. Therefore, erroneous candidates are removed from the dataset and not considered in further processing.

To prepare the training set, we need to evaluate if each candidate is correct or incorrect. Manual assessment is very cumbersome and inefficient from a time perspective, as there are 20 candidates for each question. Hence, the process needs automation. We consider a candidate correct if the results of the candidate's execution completely match the results of the query from the gold standard. However, there are several problems here. Firstly, the random coincidence: For example, a question about the headquarters of Scotland Yard has a correct answer "London". Nevertheless, the same answer can be obtained while executing the query for the question "What is the capital of Great Britain?". In this case, the incorrect candidate will be evaluated as a correct one.

Secondly, the changes in Wikidata can lead to misunderstanding, e.g., a question about the grandchildren. The query was correctly formulated as "find all grandchildren of a person". However, now there are no grandchildren of this person in Wikidata, only children of children, and the candidate is again erroneously evaluated as incorrect.

The manual test is very demanding, as there can be tens to hundreds of thousands of candidates. One of the ways out is to manually check only those candidates that the model predicted as "incorrect". It often turns out that it

[16] https://backend.app.qanswer.ai/swagger-ui/index.html

is not the model's error, but the automatic assessment of the candidate that erroneously put it correct/incorrect.

5 Evaluation and Analysis

5.1 S_1 – Classification

When working with multilinguality data, 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 [37]. 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, 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 of the 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[17]).

We determine the best-performing model while focusing on the average P@1 of a particular model over all languages. Hence, the joint objective is to achieve the highest average P@1 and the lowest standard deviation of the P@1 values. If no single model satisfies both objectives, then we build a set of all Pareto-efficient solutions.

Given our evaluation results, the set of Pareto efficient solutions is as follows $P^* = \{\text{DistilBERT},\ \text{GPT-4}\}$. This is determined by the fact that DistilBERT dominates other models on a standard deviation of F1 score objective, and GPT-4 dominates other models on average F1 score objective. Hence, there is no single model that dominates other models on both objectives.

While analyzing the results between the different model groups, MG_2 has significantly worse quality than MG_1 and MG_3 according to both metrics. While analyzing the results between the different models, GPT-4 results mostly outperform other ones for the MemQA (cf. Table 1), despite Bashkir, there is no model dominating for QAnswer for average results (cf. Table 2). Comparing P@1 results of QAnswer (cut to one query candidate as well as an average value) with each other (cf. Figure 8) shows that all models have

[17] https://purl.org/query-candidate-filtering/s1-evaluation

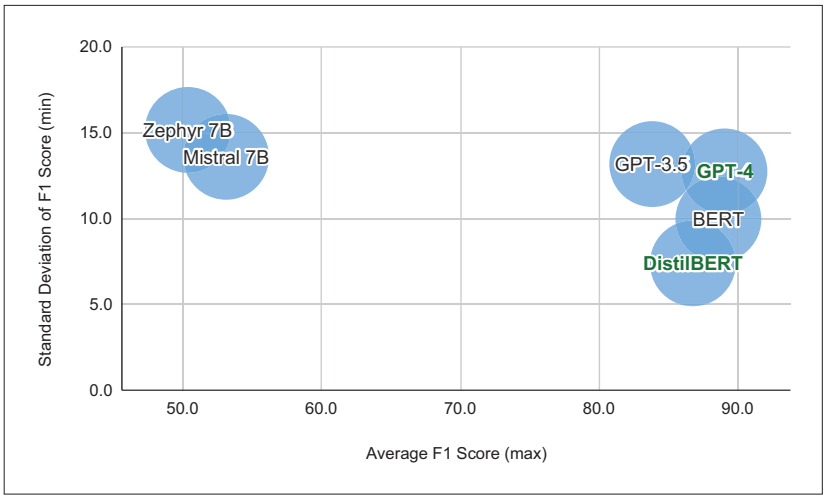

Figure 7 Two objectives of the multilingual classification efficiency visualized. The set of Pareto efficient solutions is highlighted in green.

comparable quality in results with "list of one candidate", despite ZEPHYR-7B for Spanish (no improvement). In turn, MG_1 and MG_3 have comparable quality. However, GPT-4 achieves equally high P@1 on most of the languages, except for Spanish, French, and 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 outperform open-source LLMs on our binary classification task setting (cf. Table 1).

5.2 S_2 – Question Answering

In this subsection, we present the evaluation results of the two QA systems, MemQA and QAnswer.[18] 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.

[18]The raw data is available in the online appendix at https://purl.org/query-candidate-filtering

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**

In Table 1 we present the results for ATS@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-resource languages (Belarusian, 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 Figure 8 show that 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.

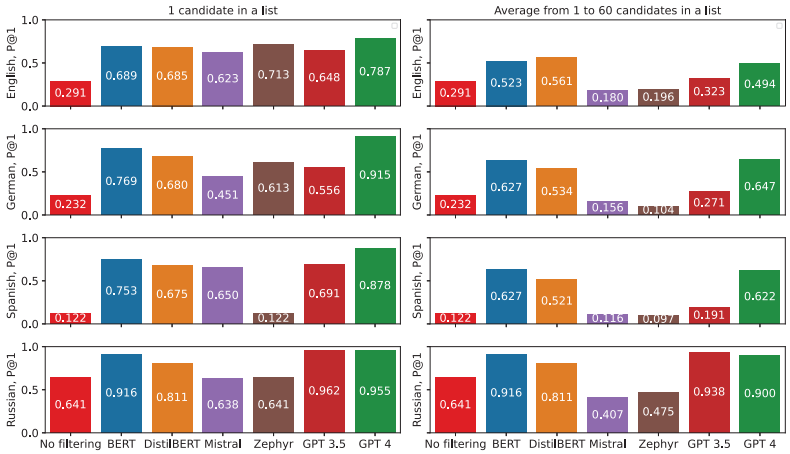

Figure 8 Precision@1 values for the QAnswer system. The left-hand side of the figure demonstrates the results when the lists of query candidates are cut off to 1 (i.e., there is no second candidate that might move to the top of the list), while the right-hand side shows average values for the candidates' 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.

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 three 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?" (English: "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").

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 Section 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

In Table 2, we present the average results for Answer Trustworthiness Score (ATS) achieved on *QAnswer*.

Figure 8 demonstrates the Precision@1 values for datasets and models before and after applying SPARQL query filtering.

6 Discussion

Our results show the strong impact of our approach on the questions in all languages. However, there are some outliers in the improved approach w.r.t. 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. [35]). 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 straightforward 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 article.

7 Conclusions and Future Work

In this article, 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 in trustworthiness as for English.
(3) Both LLMs and smaller language models can be used for our approach. Hence, users have the choice of which technology they use regarding (energy) efficiency, time investment, and costs. Our experiments show a strong quality improvement for two out of the three language model 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 language-specific prompts, as well as LLM-specific prompts. Our approach could be extended by using

additional KG properties that consider different meanings of entities separately (cf. [39]), classifications of the expected response type (e.g., [12, 33, 34]) and also consider natural-language representation of the SPARQL query (as in [39]) to improve the overall quality (cf. [40]) of very many KGQA systems. Additionally, a promising direction for improving the results would be to solve the problem of labels' non-availability of the resources. Furthermore, it seems to be reasonable to transform our approach into a pluggable component (e.g., via a question answering framework like Qanary [2, 3, 36]), s.t., many systems can benefit from the provided functionality with low time investment.

References

[1] Kushagra Singh Bisen, Sara Assefa Alemayehu, Pierre Maret, Alexandra Creighton, Rachel Gorman, Bushra Kundi, Thumeka Mgwgwi, Fabrice Muhlenbach, Serban Dinca-Panaitescu, and Christo El Morr. *Evaluation of Search Methods on Community Documents*, pages 39–49. Metadata and Semantic Research. Springer Nature Switzerland, 2023.

[2] Andreas Both, Dennis Diefenbach, Kuldeep Singh, Saedeeh Shekarpour, Didier Cherix, and Christoph Lange. Qanary – a methodology for vocabulary-driven open question answering systems. In *The Semantic Web. Latest Advances and New Domains*, pages 625–641, Cham, 2016. Springer International Publishing.

[3] Andreas Both, Kuldeep Singh, Dennis Diefenbach, and Ioanna Lytra. Rapid engineering of QA systems using the light-weight Qanary architecture. In Jordi Cabot, Roberto De Virgilio, and Riccardo Torlone, editors, *Web Engineering*, pages 544–548, Cham, 2017. Springer International Publishing.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. DeepPavlov: Open-source library for dialogue

systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127. Association for Computational Linguistics, 2018.

[6] Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. Multilingual compositional Wikidata questions. *arXiv preprint arXiv:2108.03509*, 2021.

[7] Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. Compositional generalization in multilingual semantic parsing over Wikidata. *Transactions of the ACL*, 10, 2022.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[9] Dennis Diefenbach, Andreas Both, Kuldeep Singh, and Pierre Maret. Towards a question answering system over the semantic web. *Semantic Web*, 11:421–439, 2020.

[10] Dennis Diefenbach, José Giménez-García, Andreas Both, Kamal Singh, and Pierre Maret. QAnswer KG: designing a portable question answering system over RDF data. In *European Semantic Web Conference*, pages 429–445. Springer, 2020.

[11] Pavel Efimov, Leonid Boytsov, Elena Arslanova, and Pavel Braslavski. The impact of cross-lingual adjustment of contextual word representations on zero-shot transfer. In *European Conference on Information Retrieval*, pages 51–67. Springer, 2023.

[12] Aleksandr Gashkov, Aleksandr Perevalov, Maria Eltsova, and Andreas Both. Improving the question answering quality using answer candidate filtering based on natural-language features. In *2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 635–642. IEEE, 2021.

[13] Aleksandr Gashkov, Aleksandr Perevalov, Maria Eltsova, and Andreas Both. Improving question answering quality through language feature-based SPARQL query candidate validation. In *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 2022.

[14] James Hadley. Indirect translation and discursive identity: Proposing the concatenation effect hypothesis. *Translation Studies*, 10(2):183–197, 2017.

[15] Nivas Jayaseelan. LLaMA 2: The new open source language model, 2023. https://www.e2enetworks.com/blog/llama-2-the-new-open-source-language-model.

[16] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[17] Haemin Jung and Wooju Kim. Automated conversion from natural language query to SPARQL query. *Journal of Intelligent Information Systems*, 55(3):501–520, 2020.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, Conference Track Proceedings*, 2015.

[19] Anis Koubaa. GPT-4 vs. GPT-3.5: A concise showdown. *Preprints*, March 2023.

[20] Clifford E Landers. Literary translation: A practical guide. *Multilingual Matters*, 2001.

[21] Gwénolé Lecorvé, Morgan Veyret, Quentin Brabant, and Lina M. Rojas Barahona. SPARQL-to-text question generation for knowledge-based conversational applications. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 131–147. Association for Computational Linguistics, 2022.

[22] Ekaterina Loginova, Stalin Varanasi, and Günter Neumann. Towards end-to-end multilingual question answering. *Information Systems Frontiers*, 23:227–241, 2021.

[23] T. R. McIntosh, T. Liu, T. Susnjak, P. Watters, A. Ng, and M. N. Halgamuge. A culturally sensitive test to evaluate nuanced GPT hallucination. *IEEE Transactions on Artificial Intelligence*, 1(01):1–13, 2023.

[24] Nick McKenna and Priyanka Sen. KGQA without retraining. In *ACL 2023 Workshop on SustaiNLP*, 2023.

[25] Michalis Mountantonakis, Michalis Bastakis, Loukas Mertzanis, and Yannis Tzitzikas. Tiresias: Bilingual question answering over DBpedia. In *Workshop on Deep Learning for Knowledge Graphs (DL4KG 2022)*, 2022.

[26] Diego Moussallem, Dwaraknath Gnaneshwar, Thiago Castro Ferreira, and Axel-Cyrille Ngonga Ngomo. NABU–multilingual graph-based neural RDF verbalizer. In *International Semantic Web Conference*, pages 420–437. Springer, 2020.

[27] Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. Sorry, I don't speak SPARQL: translating SPARQL queries into natural language. In *Proceedings of the 22nd international conference on World Wide Web*, pages 977–988, 2013.

[28] Axel-Cyrille Ngonga Ngomo, Diego Moussallem, and Lorenz Bühmann. A holistic natural language generation framework for the semantic web. In Ruslan Mitkov and Galia Angelova, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 819–828, Varna, Bulgaria, 2019. INCOMA Ltd.

[29] Peggy Nzomo, Isola Ajiferuke, Liwen Vaughan, and Pamela McKenzie. Multilingual information retrieval & use: Perceptions and practices amongst bi/multilingual academic users. *The Journal of Academic Librarianship*, 42(5):495–502, 2016.

[30] OpenAI. Introducing ChatGPT, 2022. https://openai.com/blog/chatGPT.

[31] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[32] Thomas Pellissier Tanon, Marcos Dias de Assunção, Eddy Caron, and Fabian M. Suchanek. Demoing Platypus – a multilingual question answering platform for Wikidata. In *The Semantic Web: ESWC 2018 Satellite Events*, pages 111–116. Springer, 2018.

[33] Aleksandr Perevalov and Andreas Both. Augmentation-based answer type classification of the SMART dataset. In Nandana Mihindukulasooriya, Mohnish Dubey, Alfio Gliozzo, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, and Ricardo Usbeck, editors, *Proceedings of the SeMantic AnsweR Type prediction task (SMART) at ISWC 2020 Semantic Web Challenge co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual Conference, November 5th, 2020*, volume 2774 of *CEUR Workshop Proceedings*, pages 1–9. CEUR-WS.org, 2020.

[34] Aleksandr Perevalov and Andreas Both. Improving answer type classification quality through combined question answering datasets. In *Knowledge Science, Engineering and Management: 14th International*

Conference, KSEM 2021, Tokyo, Japan, August 14–16, 2021, Proceedings, Part II, pages 191–204, Berlin, Heidelberg, 2021. Springer-Verlag.

[35] Aleksandr Perevalov, Andreas Both, Dennis Diefenbach, and Axel-Cyrille Ngonga Ngomo. Can machine translation be a reasonable alternative for multilingual question answering systems over knowledge graphs? In *ACM Web Conference 2022*, WWW '22. ACM, 2022.

[36] Aleksandr Perevalov, Andreas Both, Florian Gudat, Paul Bräuning, Johannes Meesters, Lennart Gründel, Marie-susann Bachmann, and Salem Zin Iden Naser. Qanary Builder: Addressing the reproducibility crisis in question answering over knowledge graphs. In *International Semantic Web Conference (ISWC) – Posters and Demos Track*, 2023.

[37] Aleksandr Perevalov, Andreas Both, and Axel-Cyrille Ngonga Ngomo. Multilingual question answering systems for knowledge graphs—a survey. *Semantic Web Journal*, 2023.

[38] Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 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.

[39] Aleksandr Perevalov, Aleksandr Gashkov, Maria Eltsova, and Andreas Both. Understanding SPARQL queries: Are we already there? Multilingual natural language generation based on SPARQL queries and large language models. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Proceedings, Part II*, volume 15232 of *Lecture Notes in Computer Science*, pages 173–191. Springer, 2024.

[40] Aleksandr Perevalov, Xi Yan, Liubov Kovriguina, Longquan Jiang, Andreas Both, and Ricardo Usbeck. Knowledge graph question answering leaderboard: A community resource to prevent a replication crisis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, pages 2998–3007, Marseille, France, 2022. European Language Resources Association.

[41] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, 2019.

[42] Anton Razzhigaev, Mikhail Salnikov, Valentin Malykh, Pavel Braslavski, and Alexander Panchenko. A system for answering simple questions in multiple languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*

(Volume 3: System Demonstrations), pages 524–537. Association for Computational Linguistics, 2023.

[43] Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. RuBQ 2.0: An innovated Russian question answering dataset. In *The Semantic Web: 18th International Conference, ESWC 2021*, pages 532–547. Springer, 2021.

[44] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[45] Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. Question answering over temporal knowledge graphs. *arXiv preprint arXiv:2106.01515*, 2021.

[46] Priyanka Sen, Alham Fikri Aji, and Amir Saffari. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *29th International Conference on Computational Linguistics*, pages 1604–1619, 2022.

[47] Li Si, Qiuyu Pan, and Xiaozhe Zhuang. An empirical analysis of user behaviour on multilingual information retrieval. *The Electronic Library*, 35(3):410–426, 2017.

[48] Lucia Siciliani, Pierpaolo Basile, Pasquale Lops, and Giovanni Semeraro. MQALD: Evaluating the impact of modifiers in question answering over knowledge graphs. *Semantic Web*, 13(2), 2022.

[49] Javier Soruco, Diego Collarana, Andreas Both, and Ricardo Usbeck. *QALD-9-ES: A Spanish Dataset for Question Answering Systems*, pages 38–52. Studies on the Semantic Web. IOS Press BV, 2023.

[50] Nikit Srivastava, Aleksandr Perevalov, Denis Kuchelev, Diego Moussallem, Axel-Cyrille Ngonga Ngomo, and Andreas Both. Lingua franca – entity-aware machine translation approach for question answering over knowledge graphs. In *Knowledge Capture Conference*. ACM, 2023.

[51] Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*, 2024.

[52] UNESCO. *Recommendation on the Legal Protection of Translators and Translations and the Practical Means to Improve the Status of*

Translators: Adopted by the General Conference at Its Nineteenth Session, Nairobi, 22 November 1976. UNESCO, 1976.

[53] Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 9th challenge on question answering over linked data (QALD-9). In *Semdeep/NLIWoD@ISWC*, 2018.

[54] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1133–1143. International World Wide Web Conferences Steering Committee, 2015.

[55] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics, 2020.

[56] Silei Xu, Theo Culhane, Meng-Hsi Wu, Sina J Semnani, and Monica S Lam. Complementing GPT-3 with few-shot sequence-to-sequence semantic parsing over Wikidata. *arXiv preprint arXiv:2305.14202*, 2023.

[57] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, Jie Zhou, Siming Chen, Tao Gui, Qi Zhang, and Xuanjing Huang. A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023.

[58] Chen Zhang, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. A review of deep learning in question answering over knowledge bases. *AI Open*, 2:205–215, 2021.

[59] Yucheng Zhou, Xiubo Geng, Tao Shen, Wenqiang Zhang, and Daxin Jiang. Improving zero-shot cross-lingual transfer for multilingual question answering over knowledge graph. In *NAACL: Human Language Technologies*, pages 5822–5834. ACL, 2021.

Biographies

Aleksandr Pervalov is a final-year Ph.D. student at the Leipzig University of Applied Sciences and the Paderborn University (both in Germany). He also leads the research project "Language Agnostic Semantic Search over Knowledge Graphs" in collaboration with Springer Nature as an industry partner. His research interests focus on applied conversational AI and question answering.

Aleksandr Gashkov is an independent researcher at the Web & Software Engineering research group, with a Ph.D. in Computational Linguistics. His research interests encompass natural language processing, multilingual question answering – both general and over knowledge graphs – and applied artificial intelligence.

Maria Eltsova is a free researcher at Web & Software Engineering research group. After promotion in linguistics in 2006, she was a professor at Perm National Research Polytechnic University (Russia) till 2022. Her research interests include computational linguistics, psycholinguistics, question answering, and digitalization of endangered indigenous languages.

Andreas Both is Head of Research at DATEV eG (a top-tier business software provider in Germany) and a professor at the Leipzig University of Applied Sciences (Germany) where he leads the Web & Software Engineering (WSE) research group which focuses on high-quality, multilingual, KG-agnostic AI methods, in particular on question answering systems, and applied AI, data, and web technologies.