Understanding SPARQL Queries: Are We Already There? Multilingual Natural Language Generation Based on SPARQL Queries and Large Language Models

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

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

Abstract. SPARQL is a standard query language for RDF data. Interpreting SPARQL queries might be a challenge, in particular, while being not familiar with the technical specifications of SPARQL or the meaning of the thing identified by a resource. In our study, we take an initial step toward employing Large Language Models to verbalize SPARQL queries, i.e., convert them to natural language. While other research often focused only on English verbalizations, we also implemented the transformation into German and Russian textual representations. The experimental framework leverages a combination of proprietary and open-source models, with enhancements achieved through further fine-tuning these models. Our methodology is assessed using the well-known question answering datasets QALD-9-plus and QALD-10, focusing on the aforementioned three languages: English, German, and Russian. To analyze performance quality, we employ metrics for machine translation alongside a survey for human evaluation. Although we encountered specific error types such as question over-specification, linguistic discrepancies, and semantic mismatches, the findings of our research indicate that Large Language Models are well-suited for the task of translating SPARQL queries into natural language, s.t., the semantics of SPARQL queries is represented in the mother tongue of the users.

Keywords: Natural Language Generation · Large Language Models · SPARQL2NL · Multilinguality · RDF2NL

1 Introduction

Current approaches for text generation from complex real-world structured data, such as Knowledge Graphs (KGs), often face the problem of semantic correctness. In particular, when considering RDF[1] KGs, converting queries written in

[1] Resource Description Framework, cf. https://www.w3.org/RDF/

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15232, pp. 173–191, 2025.
https://doi.org/10.1007/978-3-031-77850-6_10

SPARQL to natural language (i.e., *verbalizing* it) may be applied in many important scenarios. For instance, despite SPARQL being a powerful tool for users with the required technical (and domain) expertise, it still remains challenging for novice or non-technical users, especially when it comes to comprehending query semantics or while working with KGs that use resource URIs[2] that are not directly interpretable by humans (e.g., in Wikidata[3] [29]: `wd:Q937`[4] equals the physicist Albert Einstein). This challenge is partially covered by semantic parsing-based Question Answering over Knowledge Graphs (KGQA) – such systems convert a natural-language (NL) question to a SPARQL query to retrieve the answer of the user's *information need* [21]. Here, a user of a KGQA system is not required to know SPARQL at all, however, such systems have limited abilities in terms of answer quality and mostly fail to cover very complex information needs (e.g., involving aggregation, sub-queries, non-trivial property paths) [23]. Another way to address the challenge of a better understanding of SPARQL is to make the process of writing queries more transparent by converting a written query back to a natural language enabling users to compare their intention with the encoded semantics of the written SPARQL query. Such an approach is highly relevant when it comes to *explainability and trustworthiness*: a SPARQL user may get a better understanding of the information need fulfilled by a query when working with its NL verbalization.

In this work, we present an approach for converting SPARQL queries to English, German, and Russian NL representations. The ultimate goal of our approach is to provide the end users with better explainability and transparency when working with SPARQL queries. In contrast to previous studies, we focus on leveraging Large Language Models (LLMs) for NL generation and use a knowledge injection method. In addition, we conduct *experiments on multiple languages* demonstrating the quality different of LLMs when dealing with multiple languages and mitigating bias towards English-only research results. Speaking specifically about SPARQL query verbalizations, we derive the following types: *high-level* and *low-level*. The low-level verbalizations are aimed at users with proficiency in SPARQL and used for deep technical understanding of a query with the means of NL and use *technical terms* (URI, subclass of, modifier, etc.). In its turn, the high-level verbalizations are aimed at users that have no or very little knowledge about SPARQL and are represented with general-domain NL having no or very few technical terms. *Our approach is aimed at creating the high-level verbalizations* which also can be referred to as *reverse semantic parsing task*. In our study, we assessed our method using the renowned KGQA datasets, QALD-9-plus [22] and QALD-10 [28], focusing on three diverse languages: English, German, and Russian. To gauge the quality of our approach, we combined automatic metrics for machine translation and edit distance (e.g., sentence BLEU and NIST, Rouge L, Levenshtein distance) and human semantic assessments through a survey. Although we encountered various types of errors,

[2] Uniform Resource Identifiers.

[3] https://www.wikidata.org/

[4] Using `PREFIX wd:` <https://www.wikidata.org/wiki/>

the findings from our research indicate that LLMs combined with knowledge injection are well-suited for the job of transforming SPARQL queries into natural language. Hence, while in the original semantic parsing task, the goal of a system is to produce a SPARQL query given a NL question, we solve a *reverse semantic parsing task* – formulating a NL question based on a SPARQL query.

This paper aims to answer the following *research questions*:

RQ1. Is it possible to generate SPARQL query verbalizations using LLMs and knowledge injection?
RQ2. How to measure the quality of the generated verbalizations?
RQ3. What error classes are contained in the generated results?
RQ4. Do the SPARQL query features affect the NL generation quality?

This paper is structured as follows. In Sect. 2, we summarize related work on converting SPARQL to natural language. Thereafter, in Sect. 3, we present an overview of the approach proposed by us. The experimental setup is described in Sect. 4 which is followed by the analysis in Sect. 5. Finally, we discuss and conclude our work in Sect. 6. *Reproducibility statement*: The source code and data for experiments are available on GitHub[5].

2 Related Work

The previous work on the topic of conversion of SPARQL queries to NL was mostly based on grammar rules and relatively small language models (LMs). The paper by Ngonga Ngomo et al. [17] presents an approach called "SPARQL2NL". The approach involves a preliminary step that standardizes the query and identifies the types of data it contains, followed by a stage where a universal form of the query is created. Afterward, a refining phase employs simplification and substitution principles to enhance the clarity of the expression. Lastly, the process concludes with a production phase that formulates the ultimate version of the query in NL.

In [18] more general objectives of verbalizing OWL and RDF vocabularies in addition to SPARQL are targeted. The proposed approach is called "LD2NL" and also follows a sequential process which contains lexicalization, single triples realization, clustering, ordering, and grouping procedures such that the resulting text looks like a full-fledged NL. The quality of the generated text was measured through a survey that included both experts and non-experts in the Semantic Web field.

The paper by Moussallem et al. [16] focuses on a similar task as the LD2NL approach. However, the implementation here is based on an encoder-decoder architecture which uses an encoder inspired by Graph Attention Networks (GANs) and a Transformer as a decoder. The proposed approach is called NABU. The authors conduct their experiments in German, Russian, and English, and evaluate the quality using the BLEU score.

[5] https://github.com/WSE-research/SPARQL-to-NL-LLMs/

The work by Lecorvé et al. [9] concentrates on creating NL questions from SPARQL queries, with a particular interest in conversational applications such as follow-up question-and-answer interactions. The authors used the pre-trained T5 [8] and BART [11] LMs with no-context and full-context prompts. The resulting questions' quality was measured automatically with METEOR [3] and BERTScore [31] as well as using manual evaluation. The findings from both automated metrics and assessments by people indicate that while simple inquiries and common SPARQL query patterns are typically well managed, more intricate queries and aspects of dialogue, such as coreferences and ellipses, continue to pose challenges.

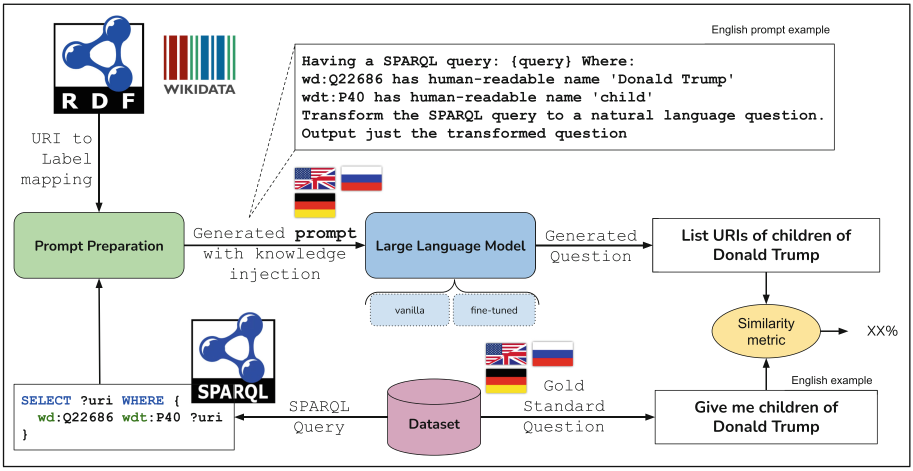
3 LLM-Driven Natural Language Generation from SPARQL

In this section, we describe our approach for the NL question generation based on SPARQL queries. The *general idea of the approach* is to leverage the abilities of LLMs for:

1. Understanding the initial information need which is encapsulated within a SPARQL query;
2. Formulating the information need as a NL question.

Therefore, we propose using instruction-tuned LLMs with prompts designed in a way to follow the *knowledge injection* method [15]. In particular, the knowledge injection is implemented as the integration of human-readable representations of URIs mentioned in a SPARQL query to a prompt. This is needed to make sure that a LLM is not dealing with unseen "anonymous" URIs (e.g., `Q937` from Wikidata). In addition, we distinguish between "vanilla" (off-the-shelf) and fine-tuned LLMs. In our approach, *we fine-tune the models based on the same prompts* with an addition of the gold-standard NL at the end. To show better generalizability of the approach, we design it in a way to work with multiple languages. Importantly, *we do not use machine translation* in the approach as it may result in an information loss, hallucinations, or named entities corruption as it was demonstrated already in [13,20].

Figure 1 demonstrates a "big picture" of our approach. Here, we first use *prompt preparation* that (1) parses a given SPARQL query from a dataset, (2) utilizes a knowledge graph (KG) for fetching the URI to label mappings (e.g., Wikidata), and (3) generates the final prompt following a pre-defined template. Thereafter, the *generated prompt is passed to a LLM* which produces a NL question intended to represent the semantics of the SPARQL query. The *generated question is then compared to a gold standard* with a particular similarity metric which has to measure the semantic meaning of both texts. It is worth mentioning that the URI to label mappings (2) and the final prompts (3) are written in a language that is to be expected from the output. For example, for the German experiments only the labels with the language tag `@de` are taken from a knowledge graph, the same is true to the prompts which are also language-specific.



```
Having a SPARQL query: {query} Where:
wd:Q22686 has human-readable name 'Donald Trump'
wdt:P40 has human-readable name 'child'
Transform the SPARQL query to a natural language question.
Output just the transformed question
```

```
SELECT ?uri WHERE {
  wd:Q22686 wdt:P40 ?uri
}
```

Fig. 1. The "big picture" of our approach for generating natural language from SPARQL queries.

The similarity metric is not necessarily to be computed in an automatic way. Hence, in our paper, we consider both automatic and human-based assessment. This allows us also to compute the correlation between the automatic metrics and the human assessment, and to get qualitative insights from the generated natural language representations. In the next section, we present a detailed experimental setup for our approach.

4 Experimental Setup

In this section, we describe datasets, prompt construction, access to LLMs, and evaluation approaches that we use to conduct our experiments.

4.1 Datasets

For our experiments, we used two datasets for evaluation and fine-tuning of the models, namely, QALD-9-plus [22] and QALD-10 [28]. Both datasets contain NL questions in multiple languages and SPARQL queries that answer the respective information needs.

The QALD-9-plus dataset is based on QALD-9 [27] which contains 558 questions and covers information from the Wikidata [29] as well as DBpedia [2] KGs. The questions are represented in eight languages: English, German, French, Russian, Ukrainian, Lithuanian, Belarus, Bashkir, and Armenian. The translations and their validation were done using the crowd-sourcing approach, where the participating crowd-workers were native speakers of the respective languages. The dataset also follows the QALD JSON structure[6].

[6] https://github.com/dice-group/gerbil/wiki/Question-Answering#web-service-interface

The QALD-10 dataset [28] introduces 402 new questions in English, Chinese, German, and Russian. The questions and SPARQL queries were written by native speakers and domain experts. The dataset also follows the QALD JSON structure.

For our experiments, we decided to focus on *three languages*: English, German, and Russian. The justification for such selection is that the authors are native speakers or fluent in at least one of them. In addition, the languages are diverse in terms of language types (analytic vs fusional) and used scripts (Latin vs Cyrillic).

4.2 Prompt Construction

In our experiments, the prompts are created based on templates for each of the considered languages. In terms of prompt engineering, we use two different settings, *zero-shot* and *one-shot*. The prompt templates for both settings contain common parts such as "head", "list", and "tail" (see Figs. 2a and 2b). While the "head" introduces a SPARQL query and the "tail" defines the instruction, the "list" contains the knowledge injection part. In particular, there were present mappings between all the mentioned URIs in a query to a human-readable representation. All SPARQL queries used for evaluation are referring to the Wikidata KG. To integrate human-readable labels, we utilize `rdfs:label` for retrieving the corresponding labels and injecting them into the prompt. Every label in the KG has to be associated with a corresponding language tag, e.g., for experiments with the Russian language: `LANG(?label) = "ru"`. Finally, the "list" part of the prompt is repeated for each of the URIs in a SPARQL query.

The one-shot setting contains an additional part which is called "shot" (see Fig. 2b). The "shot" is fulfilled recursively through the zero-shot template. However, for the "shot" also the gold-standard question is appended as an example. All the prompt parts for both settings are concatenated together in one string, following the same order as in the templates.

4.3 Access to Large Language Models

Overview. Our intention is to achieve the best possible quality, however, we also would like to compare the quality with the size (i.e., costs) of the model. Therefore, we decided to use models different in size.

Mistral-7B [7] is a 7-billion-parameter LLM. It demonstrates that a carefully designed language model is able, firstly, to deliver high performance while maintaining an efficient inference and, secondly, compress knowledge more than what was previously thought. It outperforms the previous best 13B model, LLaMA 2 [26], across all tested benchmarks. For our experiments, we use the official `Mistral-7B-Instruct-v0.2`[7] loaded in 4-bit setting. The generation is

[7] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

```
{
  "head": "Having a SPARQL query: {query} \n Where:\n ",
  "list": "{uri} has human-readable name '{uriLabel}.'",
  "tail": "\n Transform the SPARQL query to a natural language question.
  Output just the transformed question"
}
```

(a) Zero-shot prompt template example for English – the prompt is constructed sequentially based on "head", "list", and "tail".

```
{
  "shot": "--- Start Example --- \n {shot} \n --- End Example --- \n",
  "head": "Having a SPARQL query: {query} \n Where:\n ",
  "list": "{uri} has human-readable name '{uriLabel}.'",
  "tail": "\n Transform the SPARQL query to a natural language question.
  Output just the transformed question"
}
```

(b) One-shot prompt template example for English – in comparison to the zero-shot (Figure 2a) it has another part "shot" that contains a zero-shot LLM prompt as an example.

Fig. 2. Prompt templates used in our experiments

done while using the `apply_chat_template` method[8] and the `max_new_tokens`[9] parameter equal to 1/3 of the prompt size.

The GPT-3 model [5] is a 175-billion-parameter, autoregressive LLM. For all tasks, it is applied 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 [30]) showed strong performance on many NLP tasks and benchmarks in the zero-shot, one-shot, and few-shot settings. We used the `gpt-3.5-turbo` model in our experiments via the official OpenAI API[10].

The GPT-4 model released in 2023 [1] represents a large multimodal language model capable of processing image and text inputs and producing text outputs. Similarly to its previous versions, this is a transformer-based model pre-trained to predict the next token in a document. We used the `gpt-4-1106-preview` model in our experiments via the official OpenAI API.

Fine-Tuning. We fine-tuned two large language models. For Mistral-7B, the following parameters were used for the fine-tuning: `EPOCHS=2`, `BATCH_SIZE=8`, `WARMUP_STEPS=0.03`, and `LEARNING_RATE=2E-4`. The fine-tuning process was done following the PEFT method [14] with the following parameters: `LORA_ALPHA=16`, `LORA_DROPOUT=0.1`, and `TASK_TYPE=CASUAL_LM`.

[8] https://huggingface.co/docs/transformers/main/en/internal/tokenization_utils

[9] https://huggingface.co/docs/transformers/en/main_classes/pipelines

[10] https://platform.openai.com/docs/api-reference

```
"messages": [
   {
      "role":"user",
      "content":"prompt"
   },
   {
      "role":"assistant",
      "content":"gold_standard_nl"
   }
]
```

Fig. 3. A JSON format of a single fine-tuning data example.

The GPT-3.5 (`gpt-3.5-turbo`) fine-tuning was done using the OpenAI API[11] with the following parameters: `batch_size=1`, `learning_rate_multiplier=8`, and `n_epochs=3`.

For both models, we constructed a similar dataset for the fine-tuning based on prompts and gold-standard questions from the QALD-9-plus train dataset. The data was formatted following the standard "user-assistant" dialogue approach (see Fig. 3).

4.4 Evaluation and Metrics

In this section, we describe the evaluation process and metrics that we use in our experiments. Every NL question generated by a LLM is compared to a gold-standard question which is provided in the used datasets.

Automatic Metrics. For the automatic evaluation of our approach, we use machine translation and edit distance metrics, namely, Sentence BLEU [19], NIST [6] (implementation via NLTK[12]), Rouge L [12] (implementation via Python Rouge[13]), and Levenshtein Distance [10] (implementation via Python Levenshtein[14]). The aforementioned metrics are used to quantify the performance of algorithms in tasks such as translation, summarization, and other language processing applications that require comparison between generated text and reference text, therefore, they fit our task as well.

Human Semantic Evaluation. The manual human evaluation is defined as follows, we randomly selected 136 NL questions (which was the number of questions in the QALD-9-plus test dataset) from each of the used datasets

[11] https://platform.openai.com/docs/guides/fine-tuning

[12] https://www.nltk.org/

[13] https://pypi.org/project/rouge/

[14] https://pypi.org/project/python-Levenshtein/

(QALD-9-plus test, QALD-9-plus train, QALD-10). The aforementioned selection procedure was done for each of the following parameter combinations: model (e.g., Mistral-7B) and language (e.g., German). Thereafter, each of the human evaluators, who were computer science experts and either native speakers or very fluent in a given language, manually assessed the questions generated with our approach. The *human assessment* is binary: `yes` (or `true`) – when a generated NL question *semantically equivalent* to its gold standard, `no` (or `false`) – otherwise. In total, each language was covered by two human evaluators. Each evaluator had to assess the results of the five models (GPT-3.5, GPT-4, Mistral-7B, GPT-3.5 Fine-Tuned, and Mistral-7B Fine-Tuned). Based on the obtained data, we calculated an average score over the human assessments for each model-language combination.

5 Analysis

5.1 Performance of LLMs Measured with Automatic Metrics

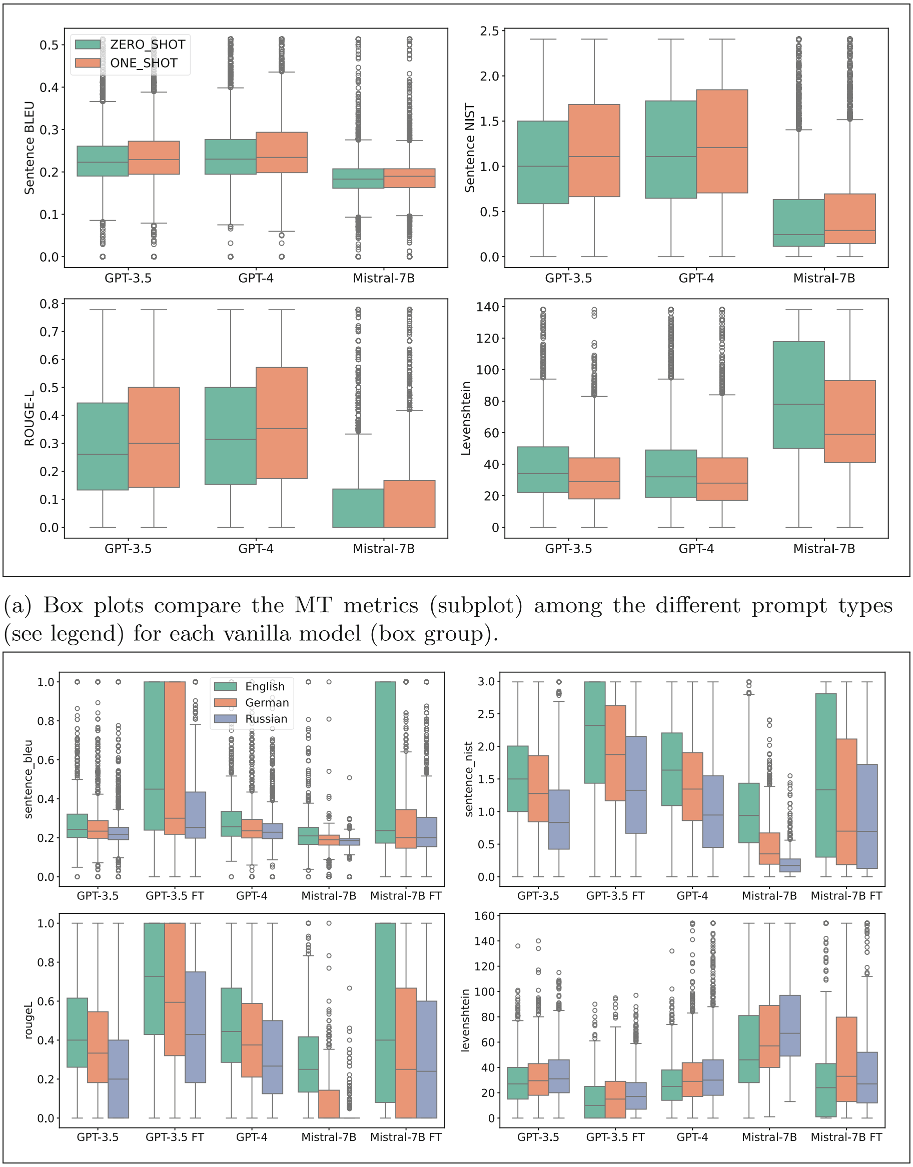
Based on the values obtained with the automatic metrics, we compared different experimental settings of our approach. In particular, what is the quality difference between zero-shot and one-shot settings, and how well the considered models perform on different languages? Regarding the Mistral-7B and GPT-3.5 models, we also compared the effect of the fine-tuning process on the quality.

Zero-Shot vs One-Shot Performance of Vanilla LLMs. Figure 4a clearly demonstrates the positive effect of using one-shot prompts in comparison to the zero-shot setting. For the models, the one-shot setting demonstrates a significant quality improvement measured by the automatic metrics. The difference for the Mistral-7B is less significant w.r.t. the one-shot setting, which may be caused by its limited capabilities in comparison to the GPT models.

Figure 4b highlights the performance of the models while looking at different languages that we considered in our experiments. The results confirm the typical assumption that the NL generation of English questions leads to a better quality than the German ones. The worst quality was achieved on the Russian questions, which may happen due to its lower presence in the NLP community as well as the different language families and used script (Cyrillic).

As both Figs. 4a and 4b suggest, the GPT-4 model outperforms the other vanilla LLMs regarding the NL generation quality. In turn, the worst generation quality is demonstrated by the Mistral-7B model. This is partially caused by the significant size difference between the considered models. Although the number of parameters for the GPT-3.5 and GPT-4 models is not public, the previous model, GPT-3, was reported as a 175 billion parameter model [25]. Hence, this is 25 times larger than the Mistral-7B model.

The Fine-Tuned vs Vanilla LLMs. As Fig. 4b shows, the fine-tuning of the both GPT-3.5 and Mistral-7B models resulted in a quality increase. In particular, the fine-tuned version of GPT-3.5 outperforms all the other models in all



(a) Box plots compare the MT metrics (subplot) among the different prompt types (see legend) for each vanilla model (box group).

(b) Box plots compare the automatic metrics between the different languages and the results of fine-tuning GPT-3.5 and Mistral-7B (see FT labels).

Fig. 4. Results of experiments

Table 1. Human evaluation accuracy scores in % for the different models split by the languages and datasets.

		GPT			**Mistral**	
		3.5	**3.5 FT**	**4**	**7B**	**7B FT**
English						
QALD-9-plus Test	🟩	68.74	71.32	**84.92**	45.21	61.03
QALD-9-plus Train	🟩	69.84	**95.21**	85.63	41.54	81.61
QALD-10	🟩	65.81	65.44	**88.59**	41.54	16.54
German						
QALD-9-plus Test	🟥	65.80	64.33	**74.99**	20.58	45.58
QALD-9-plus Train	🟥	72.79	**91.17**	85.29	4.41	68.38
QALD-10	🟥	69.85	72.06	**92.64**	5.88	11.76
Russian						
QALD-9-plus Test	🟦	54.41	49.26	**66.17**	0.00	44.85
QALD-9-plus Train	🟦	56.61	**83.82**	72.79	0.00	73.52
QALD-10	🟦	42.64	36.76	**45.58**	0.00	4.39

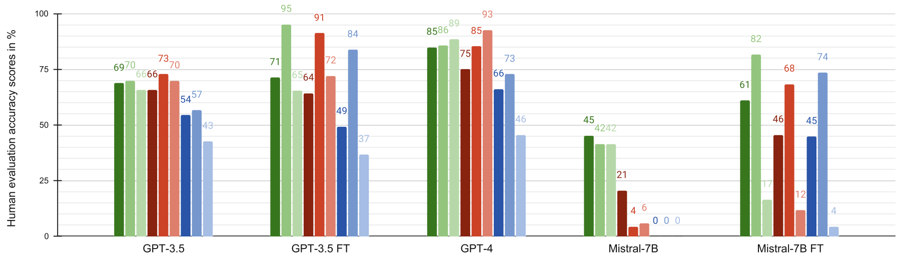


Fig. 5. Visualization of human evaluation accuracy score from Table 1.

languages. The fine-tuned Mistral-7B model significantly outperforms its vanilla version and shows results comparable to the vanilla GPT-3.5 model. Figure 4b also suggests that the variance of the metric values of the fine-tuned models is significantly higher.

5.2 Human Semantic Evaluation

The human evaluation results correspond to the fact that the given percentage of generated NL questions semantically correspond to the original gold-standard questions. In Table 1 and Fig. 5, we demonstrate the human evaluation results per model and dataset.

The values confirm the previous findings from the automated metrics regarding the vanilla model performance (GPT-4 – best quality, Mistral-7B – worst). Considering the fine-tuned models GPT-3.5 and Mistral-7B, there is an obvious quality improvement for almost all the presented settings. In particular, when applying Mistral-7B FT to German and Russian, there is a significant quality increase as the *vanilla Mistral-7B tends to answer in English despite the prompts being in the other languages*. This played a large part in the fact that the generated formulations were completely marked as incorrect by the human evaluators (cf. Table 1). The fine-tuning of Mistral-7B improves its ability to respond in multiple languages. The fine-tuned GPT-3.5 model did not achieve significant quality improvement when compared to the vanilla model. The good results of the fine-tuned models on the QALD-9-plus train split just correspond to the models' ability to memorize the training data, however, it also shows that there is no underfitting of the model.

Table 2. Correlation between the MT metrics (real values) and the human semantic evaluation (binary) when comparing questions generated out of SPARQL queries and gold-standard questions

	Sentence BLEU	Sentence NIST	Rouge L	Levenshtein
GPT-3.5				
Human Decision	0.232	0.159	0.237	−0.227
GPT-3.5 FT				
Human Decision	0.361	0.412	0.461	−0.213
GPT-4				
Human Decision	0.155	0.169	0.218	−0.187
Mistral-7B				
Human Decision	0.334	0.432	0.428	−0.364
Mistral-7B FT				
Human Decision	0.491	0.666	0.685	−0.199

While manually investigating the produced data, human evaluators mentioned that the fine-tuned models tend to generate more concise and human-natural responses, however, there are also more heavy hallucinations when the generated answer has an acceptable surface form but does not reflect the semantic meaning of its gold standard at all.

Despite the human evaluation results confirming the findings obtained on the automatic metrics, we calculated the correlation between these two different evaluation techniques. While considering the values from Table 2, one may observe that the correlation coefficients between the automatic metrics and human evaluation differ among the LLMs. This fact may correspond to the different language generation patterns of the LLMs, which from one side are captured by the human evaluation and are not captured by the automatic metrics.

5.3 Inter-language Comparison

We analyzed how the verbalization quality results differ based on the targeted natural language. Based on the median values of the box plots in Fig. 4b, we conclude that the experiments using the English language have achieved the highest results in terms of automatic metrics. The English results are followed by German and Russian respectively. The order of our results corresponds to the distribution of language coverage among the multilingual KGQA systems [21].

The human evaluation results delivered more granular insights in terms of inter-language comparison (see Table 1). While English and German experiments on commercial GPT models demonstrate comparable quality, the difference when using the Mistral model is significant and skewed towards English dominance. In addition, the human evaluation quality for Russian using the Mistral model equals zero due to the language mismatch when producing the output (see Sect. 5.4). This problem is solved only after fine-tuning the model.

5.4 Error Analysis

While conducting the human evaluation, we discovered and summarized the following error classes in the NL generation process: (1) overspecification, (2) language mismatch, and (3) semantic mismatch. An example for each of the error classes is given in Fig. 6.

Overspecification (cf. Fig. 6a). This error class represents the NL questions that contain the following drawbacks. Firstly, patterns are directly copied from a query (e.g., “child of a child” instead of “grandchild”). Secondly, occurrence specific terminology, e.g., “instance of”, “list all URIs”, etc. Finally, usage of a KG identifier in a generated NL representation, e.g., “Q1234”.

Language Mismatch (cf. Fig. 6b). This error class denotes the cases when the generated NL is in a language other than the initial prompt. For instance, prompting in German or Russian to the Mistral-7B model often leads to the results in English. Another case is when such English NL questions contain named entity labels in the original language.

```
Prompt (GPT-3.5 zero-shot):
Having a SPARQL query:
SELECT DISTINCT ?uri WHERE { wd:Q8337 wdt:P50 ?uri }
Where:
wd:Q8337 has human-readable name "Harry Potter."
wdt:P50 has human-readable name "author."
Transform the SPARQL query to a natural language question.
Output just the transformed question
Generated NL:
What is the URI of the author of the entity with the human-readable
name "Harry Potter"?
Gold Standard NL:
Who wrote Harry Potter?
```

(a) An example of the overspecification error class

```
Prompt (Mistral-7B zero-shot):
Gegeben ist die SPARQL-Anfrage:
SELECT DISTINCT ?res WHERE { wd:Q1163227 wdt:P50 ?res. }
Dabei gilt:
Die Bezeichnung von wd:Q1163227 ist "Die Säulen der Erde."
Die Bezeichnung von wdt:P50 ist "Autor."
Transformiere die SPARQL-Anfrage in eine Frage in natürlicher Sprache.
Gib nur die transformierte Frage aus.
Generated NL:
What are the distinct authors of "Die Säulen der Erde" (wd:Q1163227)?
Gold Standard NL:
Wer schrieb die Säulen der Welt?
```

(b) An example of the language mismatch error class (German prompt and English response)

```
Prompt (GPT-4 zero-shot):
Having a SPARQL query:
SELECT DISTINCT ?uri ?institute
WHERE { wd:Q76 wdt:P26/wdt:P69/wdt:P361* ?uri .
?uri wdt:P31*/wdt:P279* wd:Q3918 . }
Where:
wd:Q76 has human-readable name "Barack Obama."
wdt:P26 has human-readable name "spouse."
wdt:P69 has human-readable name "educated at."
wdt:P361 has human-readable name "part of."
wdt:P31 has human-readable name "instance of."
wdt:P279 has human-readable name "subclass of."
wd:Q3918 has human-readable name "university."
Transform the SPARQL query to a natural language question.
Output just the transformed question
Generated NL:
Which universities are either attended by Barack Obama or his spouse,
or are part of an institution that they attended?
Gold Standard NL:
What is the name of the university where Obama's wife studied?
```

(c) An example of the semantic mismatch error class

Fig. 6. Identified error classes

Semantic Mismatch (cf. Fig. 6c). This error class covers the generated questions that from one side look as full-fledged NL, however, either make no sense (i.e., hallucinated) or slightly change the original semantics, which leads to a different information need.

5.5 Analysis of SPARQL Structure and Performance

While dealing with SPARQL queries, we analyzed how their different features affect the NL generation quality. In particular, we considered the following SPARQL query features: (a) presence of `PREFIX`, (b) query type (`SELECT` or `ASK`), (c) number of triples, (d) presence of `ORDER BY`, (e) presence of `LIMIT` and `OFFSET`, and (f) presence of `HAVING` statement. All the listed query features are binary except the number of triples, however, there the maximal value is four. Therefore, we were able to analyze the automatic metrics' values while differentiating between different values of a given feature in a SPARQL query.

Based on Pearson's correlation coefficient (ρ), we identified the linear relationship between the query features and the quality metrics. The correlation analysis demonstrated that *there is a very weak relationship between the SPARQL features and automatic quality metrics*, i.e., the ρ does not exceed the absolute value of 0.14 (for Rouge-L and presence of `LIMIT` and `OFFSET` statements). Hence, this suggests that the *SPARQL structure does not affect NL generation results when using LLMs.*

6 Discussion and Conclusion

In this paper, we addressed the task of verbalizing SPARQL queries, i.e., generating NL text based on them. Using our approach, it is possible to create NL representations for public or private KGs while providing the labels of resources. Hence, there are low prerequisites for applying our approach in practice for such use cases as explainability of queries for experts or education for SPARQL beginners and many more.

While *answering RQ1*, we refer to Sects. 5.1 and 5.2. According to the evaluation values that demonstrate high-quality results in NL generation, we confirm that it is possible to generate SPARQL query verbalizations using LLMs and knowledge injection technique.

To *answer RQ2*, we refer to Sect. 5.4. Naturally, human evaluation serves as the most suitable method for measuring the quality of the generated verbalizations. As this process is expensive in every sense, one may utilize MT metrics instead. The drawback of such metrics is that they have a doubtful correlation (from very weak to moderate) with the human evaluation. Considering the error analysis, such metrics also do not recognize the listed error classes properly. Hence, there is obviously a research gap in creating such a metric that measures semantic aspects of two NL texts (cf. BERTScore [31]).

While *answering RQ3*, we also refer to Sect. 5.4. In particular, we have identified and demonstrated three error classes (1) overspecification, (2) language mismatch, and (3) semantic mismatch which have to be considered when doing further research in this direction.

Finally, while *answering RQ4*, we refer to Sect. 5.5. After analyzing the linear relationship between the automatic quality metrics and the numerical values of SPARQL query features, we found out that the query structure does not affect the text generation results.

In addition, it is worth mentioning the inter-language comparison presented in Sect. 5.3. Our findings demonstrate that the portability of the verbalization approach to non-English languages is not straightforward. In some cases, it is required to fine-tune a model to get reasonable quality results (e.g., Russian language).

Despite our approach demonstrating a successful result when applying LLMs for converting SPARQL queries to NL, it has several limitations. In particular, our approach fully depends on language-specific labels of a target KG. Moreover, each resource in a KG may have more than one label, which makes it non-trivial to decide which one to use (not always the preferred label might be the perfect choice for a verbalization). The human evaluation in this work is limited only to the English, German, and Russian languages and was done only by the authors. This obviously biases the results towards the domain-expert users. Additionally, the transferability to any other language cannot be guaranteed as the quality of the LLM as well as the language structure seem to have an impact on the quality of the natural-language verbalizations.

For future work, we will cover the aforementioned limitations and will focus on introducing better metrics for measuring the semantic meaning of a natural-language text generated from a SPARQL query. Specifically, such a metric has to prove a better correlation with human decisions. Additionally, integrating our approach into Question Answering systems (like [24]) or corresponding frameworks (like [4]) would help scientists and practitioners to understand the semantics of generated SPARQL queries.

Acknowledgments. This work was partially supported by the German Federal Ministry of Economics and Technology (BMWi) under the number 16DTM107B (*ASAGuR*) and a research project of ITZBund: "Entwicklung und Erforschung von IT-basierten Lösungen im Rahmen des ChatBot-Frameworks des Bundes (Question-Answering-Komponenten zur Erweiterung des ChatBot-Frameworks)".

References

1. Achiam, J., et al.: GPT-4 Technical report. arXiv preprint arXiv:2303.08774 (2023). https://doi.org/10.48550/arXiv.2303.08774
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52

3. Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Goldstein, J., Lavie, A., Lin, C.Y., Voss, C. (eds.) Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72. Association for Computational Linguistics, Ann Arbor, Michigan (2005). https://aclanthology.org/W05-0909
4. Both, A., Diefenbach, D., Singh, K., Shekarpour, S., Cherix, D., Lange, C.: Qanary – a methodology for vocabulary-driven open question answering systems. In: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9678, pp. 625–641. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34129-3_38
5. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901 (2020)
6. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145 (2002).https://doi.org/10.5555/1289189.1289273
7. Jiang, A.Q., et al.: Mistral 7B. arXiv preprint arXiv:2310.06825 (2023). https://doi.org/10.48550/arXiv.2310.06825
8. Kale, M., Rastogi, A.: Text-to-text pre-training for data-to-text tasks. In: Davis, B., Graham, Y., Kelleher, J., Sripada, Y. (eds.) Proceedings of the 13th International Conference on Natural Language Generation, pp. 97–102. Association for Computational Linguistics, Dublin, Ireland (2020). https://doi.org/10.18653/v1/2020.inlg-1.14, https://aclanthology.org/2020.inlg-1.14
9. Lecorvé, G., Veyret, M., Brabant, Q., Rojas Barahona, L.M.: SPARQL-to-text question generation for knowledge-based conversational applications. In: He, Y., Ji, H., Li, S., Liu, Y., Chang, C.H. (eds.) 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), pp. 131–147. Association for Computational Linguistics, Online only (2022). https://aclanthology.org/2022.aacl-main.11
10. Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710. Soviet Union (1966)
11. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.703, https://aclanthology.org/2020.acl-main.703
12. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
13. Loginova, E., Varanasi, S., Neumann, G.: Towards end-to-end multilingual question answering. Inf. Syst. Front. **23**(1), 227–241 (2021)
14. Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., Bossan, B.: PEFT: state-of-the-art parameter-efficient fine-tuning methods (2022). https://github.com/huggingface/peft
15. Martino, A., Iannelli, M., Truong, C.: Knowledge injection to counter large language model (LLM) hallucination. In: Pesquita, C., et al. (eds.) ESWC 2023. LNCS, vol. 13998, pp. 182–185. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-43458-7_34

16. Moussallem, D., Gnaneshwar, D., Castro Ferreira, T., Ngonga Ngomo, A.-C.: NABU – multilingual graph-based neural RDF verbalizer. In: Pan, J.Z., et al. (eds.) ISWC 2020. LNCS, vol. 12506, pp. 420–437. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_24
17. Ngonga Ngomo, A.C., Bühmann, L., Unger, C., Lehmann, J., Gerber, D.: Sorry, I don't speak SPARQL: translating SPARQL queries into natural language. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 977–988 (2013)
18. Ngonga Ngomo, A.C., Moussallem, D., Bühmann, L.: A holistic natural language generation framework for the semantic web. In: Mitkov, R., Angelova, G. (eds.) Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pp. 819–828. INCOMA Ltd., Varna, Bulgaria (2019). https://doi.org/10.26615/978-954-452-056-4_095, https://aclanthology.org/R19-1095
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
20. 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: Proceedings of the ACM Web Conference 2022, pp. 977–986 (2022). https://doi.org/10.1145/3485447.3511940
21. Perevalov, A., Both, A., Ngomo, A.C.N.: Multilingual question answering systems for knowledge graphs-a survey (2024). https://www.semantic-web-journal.net/system/files/swj3633.pdf. Accepted at Semantic Web Journal
22. 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). https://doi.org/10.1109/ICSC52841.2022.00045
23. Perevalov, A., Yan, X., Kovriguina, L., Jiang, L., Both, A., Usbeck, R.: Knowledge graph question answering leaderboard: a community resource to prevent a replication crisis. In: Calzolari, N., et al. (eds.) Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 2998–3007. European Language Resources Association, Marseille, France (2022). https://aclanthology.org/2022.lrec-1.321
24. Polleres, A., Diefenbach, D., Both, A., Singh, K., Maret, P.: Towards a question answering system over the Semantic Web, vol. 11, pp. 421–439. IOS Press, NLD (2020). https://doi.org/10.3233/SW-190343
25. Singh, M., Cambronero, J., Gulwani, S., Le, V., Negreanu, C., Verbruggen, G.: CodeFusion: a pre-trained diffusion model for code generation. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 11697–11708. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.716, https://aclanthology.org/2023.emnlp-main.716
26. Touvron, H., et al.: LLaMA 2: open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
27. 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)
28. Usbeck, R., et al.: QALD-10 - The 10th challenge on question answering over linked data. Semant. Web J. (2023). https://www.semantic-web-journal.net/system/files/swj3357.pdf

29. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). https://doi.org/10.1145/2629489
30. Ye, J., et al.: A comprehensive capability analysis of GPT-3 and GPT-3.5 series models. arXiv preprint arXiv:2303.10420 (2023)
31. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: evaluating text generation with Bert. arXiv preprint arXiv:1904.09675 (2019)