Towards LLM-driven Natural Language Generation based on SPARQL Queries and RDF Knowledge Graphs

Aleksandr Perevalov[1,*], Andreas Both[1,2]

[1]Leipzig University of Applied Sciences (HTWK Leipzig), Karl-Liebknecht-Straße 132, 04277 Leipzig
[2]DATEV eG, Nuremberg, Germany

Abstract

Generating natural language based on structured data has been utilized in many use cases such as data augmentation, explainability, and education. In particular, when speaking about Knowledge Graphs, one may generate natural language representation of triples (i.e., facts) or "verbalize" SPARQL queries. The latter can be treated as a reverse semantic parsing task and, for instance, can be used by non-experts to better understand the meaning of SPARQL queries, and conduct data augmentation for question answering benchmarking datasets. In this paper, we make a first attempt to utilize Large Language Models for verbalizing SPARQL queries, i.e., converting them to natural language. The experimental setup uses both commercial and open-source models and benefits from multiple prompting techniques. We evaluate our approach on the well-known question answering datasets QALD-9-plus and QALD-10 while working with three languages: English, German, and Russian. For measuring the quality, we use machine translation metrics and human evaluation (survey) together. Even though we have observed such error classes as question overspecification, language and semantic mismatch, the results of this work suggest that Large Language Models (LLMs) are a good fit for the task of converting SPARQL queries to natural language.

Keywords

SPARQL to Natural Language, SPARQL verbalization, RDF2NL, Text Generation, Large Language Models

1. Introduction

An increasing number of applications depend on RDF data and utilize the W3C SPARQL standard to query that data. Although SPARQL is a potent instrument for those with the required technical (and domain) expertise, it continues to be challenging for novice or non-technical users to comprehend the query semantics. 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* [1]. 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

3RD INTERNATIONAL WORKSHOP ON KNOWLEDGE GRAPH GENERATION FROM TEXT (TEXT2KG) Co-located with the Extended Semantic Web Conference (ESWC 2024)

*Corresponding author.
aleksandr.perevalov@htwk-leipzig.de (A. Perevalov); andreas.both@htwk-leipzig.de (A. Both)
https://perevalov.com (A. Perevalov); http://andreasboth.de (A. Both)

paths) [2]. Another way to address the challenge of better understanding of SPARQL is to make the process of writing queries more transparent by converting a written query back to a NL, i.e., *verbalizing* it. Such an approach is highly relevant when it comes to *explainability*: a SPARQL user may get a better understanding of what is query is intended to return when working with its NL verbalization.

In this work, we present an approach for converting SPARQL queries to NL. 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 knowledge injection method [3]. 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 few 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 [4] and QALD-10 [5], focusing on three languages: English, German, and Russian. To gauge the quality of our approach, we combined machine translation metrics (e.g., sentence BLEU and NIST, Rouge L, Levenshtein distance) and human assessments through survey. Although we encountered various types of errors, the findings from our research indicate that LLMs combined with knowledge injection are well-suited for the job of transforming SPARQL queries into NL.

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?

This paper is structured as follows. In Section 2, we summarize related work on converting SPARQL to NL. Thereafter, in Section 3, we present an overview of the approach proposed by us. The experimental setup is described in Section 4, which is followed by the analysis in Section 5. Finally, we discuss and conclude our work in Section 6. For the sake of reproducibility, we publish our code and data online[1].

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. [6] 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

[1] https://anonymous.4open.science/r/SPARQL-to-NL-F3B3

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 [7] 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 operations 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. [8] 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 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.

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 [10] and BART [11] LMs with no-context and full-context prompts (cf [9]). The resulting questions' quality was measured automatically with METEOR [12] and BERTScore [13] 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 based on SPARQL queries

In this section, we describe our approach for NL generation for SPARQL queries. The *general idea of the approach* is to enable LLMs to (1) "comprehend" the initial information need, which is encapsulated within a SPARQL query, and (2) to formulate the information need as a NL question. To achieve this, we propose the instruction-tuned LLMs by designing prompts that follow the knowledge injection pattern. 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., http://www.wikidata.org/entity/Q567). In addition, we distinguish between "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.

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 metric, which has to measure the semantic meaning of both texts. In the next section, we present a detailed experimental setup for our

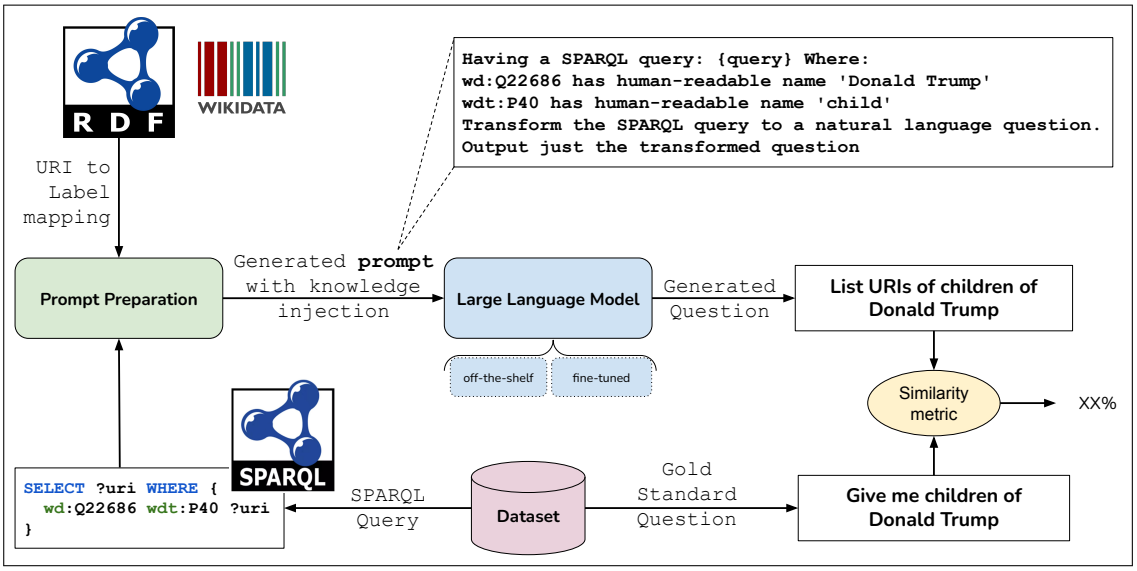

Figure 1: The "big picture" of our approach for generating natural language from SPARQL queries

approach.

4. Experimental Setup

In this section, we describe datasets, methods, and tools that we use to conduct our experiments.

4.1. Datasets

In our work, we used two datasets for evaluation, namely, QALD-9-plus [4] and QALD-10 [5]. 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 [14], which contains 558 questions and initially covers only DBpedia. It improves and extends its translations to eight languages (English, German, French, Russian, Ukrainian, Lithuanian, Belarusian, Bashkir, and Armenian), and also added the SPARQL queries for the Wikidata KG[2]. 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[3].

The QALD-10 dataset [5] 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.

[2]https://www.wikidata.org/
[3]https://github.com/dice-group/gerbil/wiki/Question-Answering#web-service-interface

```
{
  "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 – 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 – in comparison to the zero-shot (Figure 2a) it has another part "shot" that contains a zero-shot prompt as an example.

Figure 2: Prompt templates used in our experiments

4.2. Prompt Construction

In our experiments, the prompts are created based on templates for each of the considered languages. In terms of prompts 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 Figures 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. As all the SPARQL queries that we use for evaluation are for Wikidata, we utilize `rdfs:label` for retrieving the corresponding human-readable representations and putting them to the prompt. Hence, 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 Figure 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

Mistral-7B [15] 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 [16], across all tested benchmarks. For our experiments, we use the official `Mistral-7B-Instruct-v0.2`[4] loaded in 4-bit setting. In our experimental setup, we fine-tune the Mistral-7B model (Mistral-7B FT) on the training subset of the QALD-9-plus dataset. For this purpose, we use the aforementioned prompt templates that are

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

complemented with a gold-standard NL at the end. The following parameters were used for the fine tuning: `EPOCHS=2`, `BATCH_SIZE=8`, `WARMUP_STEPS=0.03`, `LEARNING_RATE=2E-4`. The fine-tuning process was done following the PEFT method[17] with the following parameters: `LORA_ALPHA=16`, `LORA_DROPOUT=0.1`, `TASK_TYPE=CASUAL_LM`.

The GPT-3 model [18] is a 175-billion-parameter, autoregressive LLM. For all tasks, it is applied without fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 (evolved to GPT-3.5 [19]) showed strong performance on many NLP tasks and benchmarks in the zero-shot, one-shot, and few-shot settings.

The GPT-4 model released in 2023 [20] 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.

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.

4.4.1. Automatic Metrics

For the automatic evaluation of our approach, we use machine translation metrics, namely, Sentence BLEU [21], NIST [22] (implementation via NLTK[5]), Rouge L [23] (implementation via Python Rouge[6]), and Levenshtein Distance [24] (implementation via Python Levenshtein[7]). 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 to our task as well.

4.4.2. Human Semantic Evaluation

The manual human evaluation is defined as follows, we randomly selected 100 NL questions from the QALD-9-plus test split for each of the following parameter combination: model (e.g., Mistral-7B) and prompt type (e.g., zero-shot). Thereafter, each of the paper authors manually compared the generated NL with the gold standard. Therefore, the *human decision* is binary. It is worth mentioning that due to limited resources within this work, we conducted the human evaluation only for the English language.

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-

[5] https://www.nltk.org/
[6] https://pypi.org/project/rouge/
[7] https://pypi.org/project/python-Levenshtein/

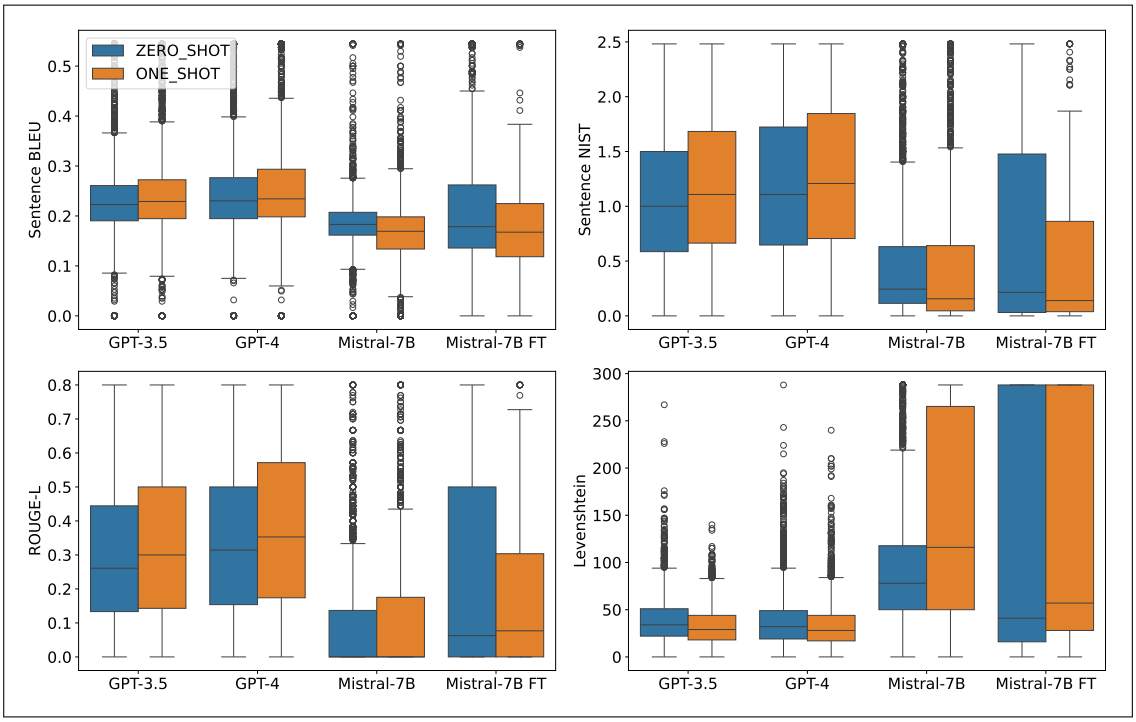

Figure 3: Box plots compare the MT metrics (subplot) among the different prompt types (see legend) for each model (box group).

shot settings, and how well the considered models perform on different languages. Regarding the Mistral-7B model, we also compared the effect of the fine-tuning process on the quality.

Figure 3 clearly demonstrates the positive effect of using one-shot prompts in comparison to the zero-shot setting. For the models except Mistral-7B, the one-shot setting demonstrates a significant quality improvement measured by the automatic metrics. The worse Mistral-7B performance w.r.t. the one-shot may be caused by its limited capabilities in comparison to the GPT models.

Figure 4 highlights the performance of the models while looking at different languages that we considered in our experiments. As we naturally assumed, 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 alphabet.

As both Figures 3 and 4 suggest, the GPT-4 model outperforms the other 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. Surprisingly, the fine-tuning of the Mistral-7B model resulted to a quality decrease. In the following section, we will refer to this situation with a qualitative analysis of the Mistral-7B FT outputs.

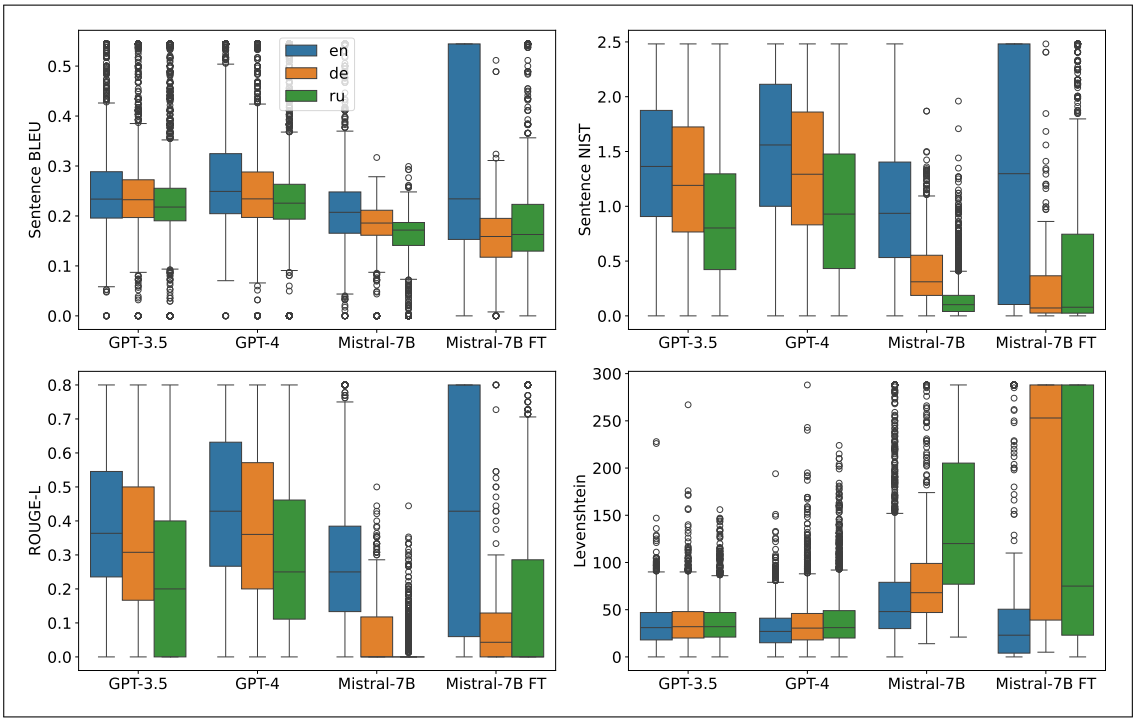

Figure 4: Box plots compare the MT metrics between the different languages

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 we demonstrate the human evaluation results per model and prompt type.

Table 1
Human evaluation accuracy scores in % for the LLMs according to the zero-shot and one-shot settings

	GPT-3.5	GPT-4	Mistral-7B	Mistral-7B FT
Zero-shot	66	83	57	42
One-shot	73	86	46	20

The values confirm the previous findings from the automated metrics regarding the model performance (GPT-4 – best quality, Mistral-7B– worst). In addition, the human evaluation scores also confirm the quality increase when using a one-shot setting for the GPT models and a reverse effect for the Mistral-7B. While manually investigating the quality decrease when applying fine-tuning procedure to the Mistral-7B model, we have identified that *the fine-tuned model produces much more "heavy" hallucinations* i.e., when the output is not even partially represents the gold-standard semantics. On the other hand, the surface form of the generated NL questions that appeared to be semantically correct was very close to the gold-standard i.e.,

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-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

the paraphrasing effect was minimal.

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. 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 Figure 5.

5.3.1. Overspecification

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”.

5.3.2. Language Mismatch

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

```
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

Figure 5: Identified error classes

5.3.3. Semantic Mismatch

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.

5.4. 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 query feature.

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). However, the aforementioned correlation coefficient represents only a linear relationship. Therefore, we decided to visualize how the quality scores differ when comparing the respective values of the query features. We identified the most significant difference when considering the (c) number of triples and (f) presence of HAVING statement. We present the corresponding visualization in Figure 6).

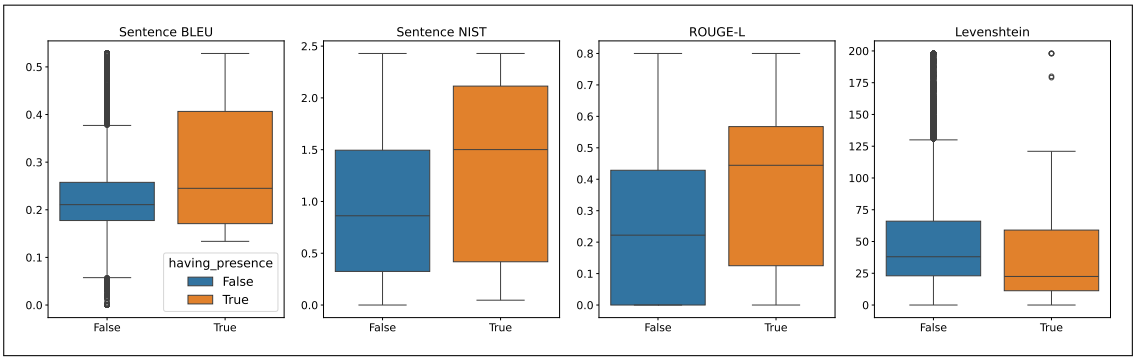

Figure 6: The "presence of HAVING statement" feature of SPARQL and its impact on the verbalization quality according to the automatic metrics

6. Discussion and Conclusion

In this paper, we addressed the task of explaining/verbalizing SPARQL queries. Using our approach, it is possible to create natural-language representations for public or private Knowledge

Graphs while providing the labels of resources. Hence, there are low prerequisites for applying our approach in practice.

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

To *answer RQ2*, we also refer to Section 5.3. 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 [13]).

Finally, while *answering RQ3*, we also refer to Section 5.3. 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.

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 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 is a perfect choice). The human evaluation in this work is limited only to the English language and was done only by the authors. This obviously biases the results towards the domain-expert users.

For future work, we will cover the aforementioned limitations and will focus on introducing better metrics for measuring the semantic meaning of a NL text generated based on a SPARQL query. Specifically, such a metric has to prove a better correlation with human decisions.

References

[1] A. Perevalov, A. Both, A.-C. N. Ngomo, Multilingual question answering systems for knowledge graphs—a survey (2024). URL: https://www.semantic-web-journal.net/system/files/swj3633.pdf, under review.

[2] A. Perevalov, X. Yan, L. Kovriguina, L. Jiang, A. Both, R. Usbeck, Knowledge graph question answering leaderboard: A community resource to prevent a replication crisis, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 2998–3007. URL: https://aclanthology.org/2022.lrec-1.321.

[3] A. Martino, M. Iannelli, C. Truong, Knowledge injection to counter large language model (llm) hallucination, in: C. Pesquita, H. Skaf-Molli, V. Efthymiou, S. Kirrane, A. Ngonga, D. Collarana, R. Cerqueira, M. Alam, C. Trojahn, S. Hertling (Eds.), The Semantic Web: ESWC 2023 Satellite Events, Springer Nature Switzerland, Cham, 2023, pp. 182–185.

[4] A. Perevalov, D. Diefenbach, R. Usbeck, A. Both, QALD-9-plus: A multilingual dataset for question answering over DBpedia and Wikidata translated by native speakers, in: International Conference on Semantic Computing (ICSC), 2022.

[5] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C. N. Ngomo, M. Saleem, A. Both, QALD-10 — The 10th Challenge on Question Answering over Linked Data, Semantic Web Journal (2023). URL: https://www.semantic-web-journal.net/system/files/swj3357.pdf.

[6] A.-C. Ngonga Ngomo, L. Bühmann, C. Unger, J. Lehmann, D. Gerber, Sorry, I don't speak SPARQL: translating SPARQL queries into natural language, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 977–988.

[7] A.-C. Ngonga Ngomo, D. Moussallem, L. Bühmann, A holistic natural language generation framework for the semantic web, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), INCOMA Ltd., Varna, Bulgaria, 2019, pp. 819–828. URL: https://aclanthology.org/R19-1095. doi:10.26615/978-954-452-056-4_095.

[8] D. Moussallem, D. Gnaneshwar, T. Castro Ferreira, A.-C. Ngonga Ngomo, NABU–multilingual graph-based neural RDF verbalizer, in: International Semantic Web Conference, Springer, 2020, pp. 420–437.

[9] G. Lecorvé, M. Veyret, Q. Brabant, L. M. Rojas Barahona, SPARQL-to-text question generation for knowledge-based conversational applications, in: Y. He, H. Ji, S. Li, Y. Liu, C.-H. Chang (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), Association for Computational Linguistics, Online only, 2022, pp. 131–147. URL: https://aclanthology.org/2022.aacl-main.11.

[10] M. Kale, A. Rastogi, Text-to-text pre-training for data-to-text tasks, in: B. Davis, Y. Graham, J. Kelleher, Y. Sripada (Eds.), Proceedings of the 13th International Conference on Natural Language Generation, Association for Computational Linguistics, Dublin, Ireland, 2020, pp. 97–102. URL: https://aclanthology.org/2020.inlg-1.14. doi:10.18653/v1/2020.inlg-1.14.

[11] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880. URL: https://aclanthology.org/2020.acl-main.703. doi:10.18653/v1/2020.acl-main.703.

[12] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss (Eds.), Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72. URL: https://aclanthology.org/W05-0909.

[13] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with Bert, arXiv preprint arXiv:1904.09675 (2019).

[14] R. Usbeck, R. H. Gusmita, A.-C. N. Ngomo, M. Saleem, 9th challenge on question answering over linked data (QALD-9), in: Semdeep/NLIWoD@ISWC, 2018.

[15] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao,

T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, arXiv preprint arXiv:2310.06825 (2023). doi:10.48550/arXiv.2310.06825. arXiv:2310.06825.

[16] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., LLaMA 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[17] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, PEFT: State-of-the-art parameter-efficient fine-tuning methods, https://github.com/huggingface/peft, 2022.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[19] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, et al., A comprehensive capability analysis of GPT-3 and GPT-3.5 series models, arXiv preprint arXiv:2303.10420 (2023).

[20] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., GPT-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[21] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[22] G. Doddington, Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, in: Proceedings of the second international conference on Human Language Technology Research, 2002, pp. 138–145.

[23] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.

[24] V. I. Levenshtein, et al., Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, volume 10, Soviet Union, 1966, pp. 707–710.

[25] M. Singh, J. Cambronero, S. Gulwani, V. Le, C. Negreanu, G. Verbruggen, CodeFusion: A pre-trained diffusion model for code generation, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 11697–11708. URL: https://aclanthology.org/2023.emnlp-main.716. doi:10.18653/v1/2023.emnlp-main.716.