Enhancing Answers Verbalization using Large Language Models

Daniel VOLLMERS Parth SHARMA Hamada M. ZAHERA
Axel-Cyrille NGONGA-NGOMO
[a] *Data Science Group, Paderborn University, Germany*
ORCiD ID: Daniel Vollmers https://orcid.org/0000-0002-5324-4952, Hamada M. Zahera https://orcid.org/0000-0003-0215-1278, Axel-Cyrille Ngonga-Ngomo https://orcid.org/0000-0001-7112-3516

Abstract. *Purpose*: This study investigates the verbalization of answers generated by knowledge graph question answering (KGQA) systems using large language models. In user-centric applications, such as dialogue systems and voice assistants, answer verbalization is an essential step to enhance the quality of interactions.

Methodology: We experimented with different large language models to verbalize answers from knowledge-based question-answering systems. In particular, we fine-tuned the LLM models (T5, BART and PEGASUS) on different inputs, including SPARQL queries and triples, to determine which model performs best for answer verbalization.

Findings: We found that fine-tuning language models and introducing additional knowledge such as SPARQL queries, achieve state-of-the-art results in verbalizing answers from KGQA systems.

Value: Our approach can be used to generate answers verbalization for different KGQA systems, including dialogue systems or voice assistants.

Keywords. Knowledge Graph, Verbalization, Question Answering, KBQA

1. Introduction

In recent years, knowledge graph question answering (KBQA) systems have emerged as essential access points to knowledge graphs [18]. These systems allow users to interactively find information interactively without requiring a deep understanding of the underlying structure of knowledge graphs. Current approaches of KGQA can be categorized into two types: (1) semantic parsing approaches, which generate logical queries (e.g., SPARQL), and (2) retrieval-based approaches, which extract and rank entities or literals from knowledge graphs. Both approaches generate answers in the form of single entities, sets of entities, or literal values. However, practical applications like chatbots or speech assistants often need more natural, human-like responses. Consequently, there is a growing need to verbalize answers generated by KGQA systems. By presenting answers in natural language, verbalization makes information more accessible to a wider audience, including those who may not be familiar with linked data or query languages used by KGQA systems [6, 8]. Current open-source KGQA systems typically provide answers without verbalizing them in natural language [4, 8]. This lack of verbalization

makes interaction with users less natural compared to voice assistants such as Siri and Alexa. Few studies have been proposed to address this problem. For example, VOGUE framework [7] has been designed to generate natural language explanations for visual elements in user interfaces. This allows AI systems to provide verbal descriptions and rationales for graphical components, making the interaction more conversational.

To verbalize answers in KGQA, multiple inputs can be used, including *questions*, and *answers*. KGQA systems can produce various answer types, such as single entities, lists of entities, or literals like numbers or text. Additionally, semantic parsing systems generate a logical query, which can be crucial for answer verbalization. Furthermore, it is possible to extract additional information, such as entity labels, from knowledge graphs. This wide range of input sources and answer types makes verbalizing answers for KGQA systems a challenging task. Recently, transfer learning of large language models (LLMs) has shown impressive performances in text generation tasks, including query generation for various structured and unstructured inputs [1]. In our study, we focuse on fine-tuning different LLMs, such as T5 or BART, to address the challenges of answer verbalization. Furthermore, we experiment with different inputs to assess the impact of logical form and answers on the quality of the verbalized output. We summarize the main contributions of our study as follows:

- Fine-tuning LLMs achieves significant results for answers verbalization in KGQA systems.
- Incorporating additional information in the LLM input such as logical forms or triples yields better verbalized answers.
- We provide the implementation of our approach and the datasets used in our experiments on the GitHub repository.[1]

2. Related Work

In KGQA systems, there are two main methods: semantic parsing-based and retrieval-based. Semantic parsing-based methods generate a query that can be executed on a database, while retrieval-based methods extract answers directly from the knowledge graph using retrieval and ranking techniques [10]. Furthermore, semantic parsing-based methods generate queries that provide an additional source for answer verbalization. In contrast, retrieval-based methods use only the input question and the generated answers for verbalization. This paper focuses on verbalizing answers from semantic parsing-based KGQA systems. For answer verbalization in KGQA systems, most approaches usullay consider a encoder-decoder architecture. For example, VOGUE [7] framework takes inputs from both the question and logical form through a dual encoder model. These inputs are then combined using cross-attention, and a hybrid decoder generates the final natural language sequence. Recent text generation methods use pre-trained language models like T5 [15] and BART [11] for various NLP tasks. For instance, Montella et al. [13] applied transfer learning with pre-trained models such as T5 and BART to verbalize answers from KGQA systems, using questions and answers as inputs. They also applied a masking technique to improve the generalization of test datasets. Their results indicate that transfer learning enhances the performance of answer verbalization compared to the VOGUE model.

[1] https://github.com/dice-group/QAAnswerVerbalizer

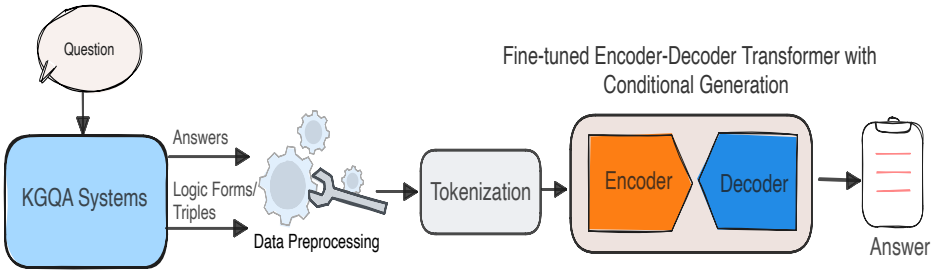

Figure 1. The architecture of our verbalization model for generating natural language answers.

3. Approach

Figure 1 shows the architecture of our approach, including a fine-tuned encoder-decoder model. Our approach takes a question, its answer, and a query from a KGQA system as an input. Further information, such as relevant triples, in included as alternative input. These inputs are preprocessed to clean noisy data and tokenized to generate the token vector for the language model. The following sections describe the details of preprocessing step and large language models used in our study.

3.1. Preprocessing Step

We conducted our experiments with two different forms of input in our verbalization model. In both versions, we included the answer set and the input question as input. Additionally, we explored using triples and SPARQL queries as additional inputs. In case of SPARQL queries, language models often encounter problems with special tokens such as *{* or *}*, that are mapped to unknown tokens then. Therefore, it is necessary to replace them with different tokens. In particular, we apply the following replacements:

- A variable such as *?uri* is replaced by the string *var_uri*
- '*{*' and '*}*' are replaced by the strings *brack_open*and *brack_close* respectively.
- The period '.' is replaced by the string *sep_dot*
- All SPARQL keywords are converted to lowercase
- All prefixes are removed.

```
Question:          What is the nationality of Aishath Saffa?

Logical Form:      select distinct var_uri where brack_open
                   Aishath_Saffa nationality var_uri brack_close

Triple(s):         Aishath Saffa nationality Maldives

Verbalization:     Maldives is the nationality of Aishath Saffa.
```

Listing 1: An example of input preprocessing

In both approaches (queries and triples), we replace the URIs of entities and relations with their labels from the knowledge graph. URIs consist of hashes or numeric identifiers

that lack semantic information. For example, we use *Aishath Saffa* string as input for our model for the DBpedia [12] entity http://dbpedia.org/resource/Aishath_Saffa. Listing 1 shows the output of the preprocessing steps for the example question *What is the nationality of Aishath Saffa?*.

3.2. Large Language models

We experimented with three different pre-trained models from the literature T5, BART, and PEGASUS, which are briefly described in the following.

T5 model [15] is a unified transformer model specifically designed for text-to-text tasks, with a focus for fine-tuning on different downstream applications. It has shown quite competitive performance across wide range of NLP tasks such as query generation [1, 19]. Moreover, T5 has been successfully applied to question answering tasks, such as query generation and entity linking, making it a good choice for answer verbalization.

BART model [11] is a popular LLM for generating natural language sequences. Similar to T5, it enables fine-tuning for various NLP-related tasks. Furthermore, BART has demonstrated strong performance in tasks involving text generation and comprehension, including applications such as entity linking [3].

PEGASUS model [20] is originally designed for generating summaries of natural language texts. Since answers verbalization from KGQA systems can be regarded as a summarization task, where various inputs such as questions, answers, and logical forms shall be summarized into a short natural language text. In our experiments, we investigated whether fine-tuning a summarization model can enhance the performance of answer verbalization. We trained two versions of LLMs: one model using triples as input and another model using queries along with answers and input questions as input.

All models were fine-tuned with an equal number of epochs and input samples[2].

4. Evaluation

In this section, we describe the set up of our experiments including datasets, baselines and evaluation metrics for answering the research questions:

- **RQ**$_1$: Does incorporating structured data from knowledge graphs, such as triples or queries, improve the performance of LLMs in answer verbalization?
- **RQ**$_2$: What types of inputs are most effective for generating natural language questions?

4.1. Datasets

VQuAnDa (Verbalization QUestion ANswering DAtaset [9]) is one of the first datasets that contains natural language verbalizations for the questions. It contains 5000 examples, the SPARQL queries (DBpedia [12]), and their verbalizations. The dataset is based on the largescale complex question-answering dataset (LC-QuAD [17]).

[2]Training setup: https://github.com/dice-group/QAAnswerVerbalizer/blob/main/args.py

Table 1. Comparison with baselines. The results for the baselines are taken from the corresponding papers. The inputs to the models: **Q** represents the question, **LF** represents Logical form/query, **T** represents Triples(s) and **H** represents Hybrid which is a combination of both LF and Q. The BLEU-4 score is reported. The scores are normalized on a scale of $[0, 100]$.

	BLEU		**METEOR**	
Model	**VQuAnDa**	**ParaQA**	**VQuAnDa**	**ParaQA**
Transformer [7] (**Q**)	18.37	23.61	56.83	59.64
Transformer [7] (**LF**)	23.18	28.01	60.17	63.75
BERT [7] (**Q**)	22.78	26.12	59.28	62.59
BERT [7] (**LF**)	26.48	30.31	65.92	65.92
VOGUE [7] (**H**)	28.76	32.05	67.21	68.85
T5 [13] (**Q**)	39.07	30.62	67.70	59.81
BART [13](**Q**)	43.90	35.57	71.92	65.40
PEGASUS **(Q+LF)**	45.97	**50.18**	79.87	80.70
PEGASUS **(Q+T)**	45.26	48.48	80.24	**81.97**
BART **(Q+LF)**	45.43	46.48	78.80	80.21
BART **(Q+T)**	43.02	47.32	78.57	80.98
T5 **(Q+LF)**	**49.25**	47.49	**80.66**	80.26
T5 **(Q+T)**	45.02	45.91	79.55	79.87

ParaQA [5] is formed using the verbalizations in VQuAnDa[9] and contains up to 8 paraphrased verbalizations of the answer on the DBpedia KG, along with the question and the SPARQL query. It contains 5000 examples.

4.2. Metrics

For evaluation, we use the well-established metrics BLEU and METEOR, for comparing our approach against the baseline methods. We briefly summarize each metric as follows:

BLEU [14] (Bilingual Evaluation Understudy): This metric compares *n*-grams between generated sentences and reference ones. The BLUE metric also includes a brevity penalty based on the lengths of reference and generated sentences, penalizing when the generated text is shorter than the reference ones. The brevity penalty is not calculated for each sentence in a corpus to avoid penalizing shorter sentences. The scores of BLEU metric ranges from 0 to 1, where 1 is the best score, indicating that the reference and generated text are identical.

METEOR [2] (Metric for Evaluation of Translation with Explicit Ordering): This metric evaluates the similarity between the hypothesis (i.e., generated text) and the reference text by considering chunks of text. Unlike other metrics, METEOR can incorporate semantic similarity, allowing it to account for similar words, not just exact matches. It also gives more weight to recall compared to precision, providing a more balanced score. METEOR calculates the Harmonic Mean of precision and recall and counts exact word matches between the hypothesis and the reference. Additionally, it penalizes incorrect word order, as the sequence of words in a sentence is essential for grammatical correctness and meaning.

4.3. Results

In this section, we present our results achieved for answering both of our research questions. All results are presented in Table 1.

Incorporating structura data from knowledge graphs in LLMs (RQ_1): The evaluation results shows that incorporating structured knowledge, such as triples and queries, enhances the performance of answer verbalization on both datasets. Existing models in the literature use fine-tuned LLMs for answer verbalization. In contrast, our approach includes triples or SPARQL queries in the verbalization process, yielding a better performance in verbalization. Other models such as VOGUE also use logical forms, but without using pre-trained LLMs such as T5 or BART, which contributes the performance margin in the results.

Experiment with different forms of input (RQ_2): Our findings indicate that verbalizing SPARQL queries rather than triples generally yields superior performance across various datasets, with the exception of the PARAQA dataset where the PEGASUS model showed slightly better results. This marginal difference suggests that queries, which include information about aggregations used, may offer an advantage over simply displaying the knowledge graph structure.

5. Conclusion

In this study, we have shown, that introducing structure information into an LM for verbalizing answers to natural language questions improves the performance in answer verbalization. Furthermore, the results have shown, that logical forms such as SPARQL queries often lead to better results compared to introducing triples. Our answer verbalization approach can be used as an extension for each question-answering model that is capable of producing logical forms such as SPARQL queries. Alternatively, triples can be applied, in cases, where no queries are available. However, there are only a few KGQA datasets available at the moment mostly on DBpedia KG, that include verbalizations, so extending existing KGQA datasets is necessary to improve verbalizations on other KGs. On the other hand, using current interaction-based LMs such as LLAMA [16] might also be an alternative for training LMs for verbalization.

Acknowledgement

This work has been supported by the European Union's Horizon Europe research and innovation programme (grant No 101070305), the German Federal Ministry of Education and Research (BMBF) within the projects KIAM (grant no 02L19C115), COLIDE (grant no 01I521005D), and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation): TRR 318/1 2021 – 438445824.

References

1. Banerjee D, Nair PA, Kaur JN, Usbeck R, Biemann C. Modern Baselines for SPARQL Semantic Parsing. In: Proceedings of the 45th International ACM SIGIR Conference

on Research and Development in Information Retrieval SIGIR '22, ACM; 2022. http://dx.doi.org/10.1145/3477495.3531841.
2. Banerjee S, Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization Ann Arbor, Michigan: Association for Computational Linguistics; 2005. p. 65–72. https://aclanthology.org/W05-0909.
3. De Cao N, Izacard G, Riedel S, Petroni F. Autoregressive Entity Retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021 OpenReview.net; 2021. https://openreview.net/forum?id=5k8F6UU39V.
4. Fu B, Qiu Y, Tang C, Li Y, Yu H, Sun J. A survey on complex question answering over knowledge base: Recent advances and challenges. arXiv preprint arXiv:200713069 2020;.
5. Kacupaj E, Banerjee B, Singh K, Lehmann J, ParaQA: A Question Answering Dataset with Paraphrase Responses for Single-Turn Conversation; 2021.
6. Kacupaj E, Premnadh S, Singh K, Lehmann J, Maleshkova M. Vogue: answer verbalization through multi-task learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases Springer; 2021. p. 563–579.
7. Kacupaj E, Premnadh S, Singh K, Lehmann J, Maleshkova M. VOGUE: Answer Verbalization through Multi-Task Learning. CoRR 2021;abs/2106.13316. https://arxiv.org/abs/2106.13316.
8. Kacupaj E, Singh K, Maleshkova M, Lehmann J. An Answer Verbalization Dataset for Conversational Question Answerings over Knowledge Graphs. arXiv preprint arXiv:220806734 2022;.
9. Kacupaj E, Zafar H, Lehmann J, Maleshkova M. VQuAnDa: Verbalization QUestion ANswering DAtaset. In: Harth A, Kirrane S, Ngonga Ngomo AC, Paulheim H, Rula A, Gentile AL, et al., editors. The Semantic Web Cham: Springer International Publishing; 2020. p. 531–547.
10. Lan Y, He G, Jiang J, Jiang J, Zhao WX, Wen JR, A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions; 2021.
11. Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, 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, editors. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics Online: Association for Computational Linguistics; 2020. p. 7871–7880. https://aclanthology.org/2020.acl-main.703.
12. Mendes P, Jakob M, Bizer C. DBpedia: A Multilingual Cross-domain Knowledge Base. In: Calzolari N, Choukri K, Declerck T, Doğan MU, Maegaard B, Mariani J, et al., editors. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12) Istanbul, Turkey: European Language Resources Association (ELRA); 2012. p. 1813–1817. http://www.lrec-conf.org/proceedings/lrec2012/pdf/570_Paper.pdf.
13. Montella S, Rojas-Barahona L, Bechet F, Heinecke J, Nasr A. Transfer Learning and Masked Generation for Answer Verbalization. In: Chen W, Chen X, Chen Z, Yao Z, Yasunaga M, Yu T, et al., editors. Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI) Seattle, USA: Association for Computational Linguistics; 2022. p. 47–54. https://aclanthology.org/2022.suki-1.6.

14. Papineni K, Roukos S, Ward T, Zhu WJ. Bleu: a Method for Automatic Evaluation of Machine Translation. In: Annual Meeting of the Association for Computational Linguistics; 2002. .
15. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research 2020;21(140):1–67. http://jmlr.org/papers/v21/20-074.html.
16. Touvron H, Lavril T, Izacard G, Martinet X, Lachaux MA, Lacroix T, et al. LLaMA: Open and Efficient Foundation Language Models. ArXiv 2023;abs/2302.13971. https://api.semanticscholar.org/CorpusID:257219404.
17. Trivedi P, Maheshwari G, Dubey M, Lehmann J. Lc-quad: A corpus for complex question answering over knowledge graphs. In: International Semantic Web Conference Springer; 2017. p. 210–218.
18. Wu P, Zhang X, Feng Z. A survey of question answering over knowledge base. In: Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding: 4th China Conference, CCKS 2019, Hangzhou, China, August 24–27, 2019, Revised Selected Papers 4 Springer; 2019. p. 86–97.
19. Ye X, Yavuz S, Hashimoto K, Zhou Y, Xiong C. RNG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering. In: Muresan S, Nakov P, Villavicencio A, editors. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) Dublin, Ireland: Association for Computational Linguistics; 2022. p. 6032–6043. https://aclanthology.org/2022.acl-long.417.
20. Zhang J, Zhao Y, Saleh M, Liu PJ, PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization; 2019.