nlp_enjoyers at Text-Graph Representations for Knowledge Graph Question Answering using all-MPNet

Nikita Kurdiukov[1], **Viktoriia Zinkovich**[1], **Pavel Tikhomirov**[1], **Sergey Karpukhin**[1]
[1]The Skolkovo Institute of Science and Technology / Moscow, Russia
{n.kurdiukov, v.zinkovich, p.tikhomirov, s.karpukhin}@skoltech.ru

Abstract

This paper presents a model for solving the Multiple Choice Question Answering (MCQA) problem, focusing on the impact of subgraph extraction from the Knowledge Graph on model performance. The proposed method combines textual and graph information by adding linearized sub-graphs directly into the main question prompt with separate tokens, improving the performance of models working with each modality separately. The study also includes an examination of Language Model (LLM) backbones and the benefit of linearized subgraphs and sequence lengt, with efficient training provided by fine-tuning with LoRA. The top benchmark with the usage of subgraphs and MPNet achieved an F1 score of 0.3887. The main limitation of the experiments is the usage of pre-generated subgraphs/triplets from the graph, and the lack of exploration of in-context learning and prompting strategies with decoder architectures.

1 Introduction

With the exponential growth of digital information, the need to develop tools for prompt and efficient data retrieval has become the most urgent in the field of Natural Language Processing (NLP). Many state-of-the-art approaches have been proposed to solve such problems, especially encoder-only models, including BERT (Devlin et al., 2019) and its variants such as RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2020), that show good performance in the retrieval task.

However, one important area of research focuses on solving Multiple Choice Question Answering problems (MCQA), when the model needs to select one correct answer among several options autonomously without external context (Huang et al., 2022). This task remains quite challenging in NLP, as in order to answer a quiz question, the model to be developed should not only have a large knowledge base (Talmor et al., 2019), but also be able to make logical inferences (Li et al., 2022).

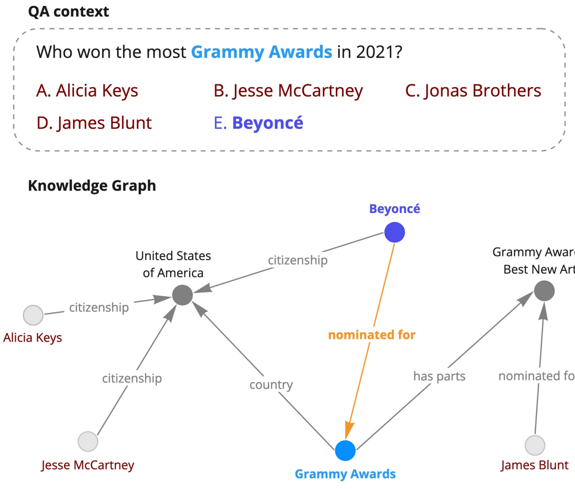


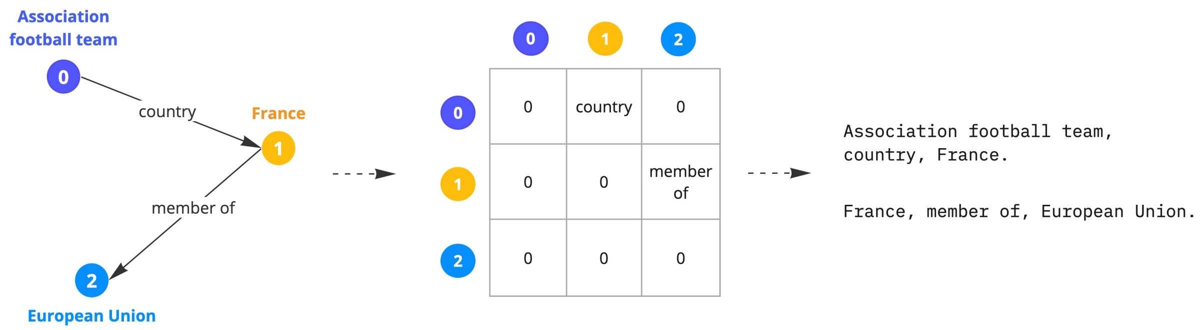
Figure 1: Knowledge Graph example

To solve such tasks different LLMs can be applied, e.g. T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) which are encoder-decoder models for natural language generation (NLG). However even such SOTA models can generally fall short on MCQA. One popular reason is that models try to predict the most likely answer in terms of grammatical construction of a sentence without considering the logical coherence of the text (Robinson et al., 2023).

To enhance the performance of LLM, in the following work, we incorporate structured knowledge graphs into the model training process (Fig.1), as this method has been noted many times in earlier works (Salnikov et al., 2023). The graph is obtained by taking the shortest paths from all mentioned concepts in corresponding questions to a candidate answer entity in the knowledge graph of Wikidata.

Thus, the **main contribution of the following work** are as follows:

- We propose a method of combining textual and graph information. Adding linearized sub-



	0	1	2
0	0	country	0
1	0	0	member of
2	0	0	0

```
Association football team,
country, France.

France, member of, European Union.
```

Figure 2: Example of the process of a subgraph linearization into text

graph directly into main question prompt with additional separate tokens allows to improve the performance of models working with each modality separately.

- We conducted a thorough study of LLM backbones and wide hyper-parameter search. For efficient training we provided fine-tuning with LoRA.

2 Method

We propose implementing the MPNet (Song et al., 2020) model, training it on question-answer pairs with incorporated linearized knowledge graphs. Additionally, we utilize the LoRA implementation from the `peft` library and apply oversampling techniques to address imbalance in the training dataset.

Our approach ultimately relies on tuning of LLM for binary classification task while also including information from wiki-data graph domain in LLM pipeline. The representations for target prediction on question-answer pair is acquired by addressing the last hidden layer representation of the `[CLS]` token of the model.

According to the nature of task it is obvious that amongst candidate answers only one of them is correct, however the amount of the candidate answers for single question is not known beforehand. During inference we utilize knowledge about only one candidate answer being right and select the most probable answer to be correct according to model scores. This naturally allows to use model trained for classification target for ranking top-1 candidate answer.

2.1 Dataset

For our research, we utilized the TextGraphs17-shared-task dataset, consisting of 37,672 question-answer pairs annotated with Wikidata entities. This dataset includes 10 different types of data, notably entities from Wikidata mentioned in both the answer and the corresponding question, as well as a shortest-path graph for each `<question, candidate answer>` pair.

2.2 Evaluation metrics

During training and evaluation of our models we use metrics same as ones present in the workshop leaderboard, which include **accuracy, precision, recall** and **F1-score**. It is important to note that accuracy here is quite uninformative due to the dataset's imbalance, with incorrect answers constituting 90% of the data.

2.3 Input preprocessing

Since the subgraphs from the knowledge graph are already provided, it is left to preprocess them for the model. In order to incorporate information from the subgraphs, the later are linearized into a text according to Salnikov et al. (2023). The process is nearly identical, except for separating distinct triplets with a semicolon. To elaborate, subgraphs are converted to a binary adjacency matrix. If nodes indexed i and j are connected, their edge label is stored in the corresponding [i, j] matrix element. The matrix is then unraveled row by row to generate linearized sentences from corresponding triples (`node_from`, edge, `node_to`) in the adjacency matrix (Fig.2).

The resulting input text for the model has the following form: `Question entities + ' : ' + Question + ' [SEP] ' + Linearized graph`. Details of various backbones, processing pipelines and scores are reported in following Sections 3 and 4.

Model	F1 Score
T5-Small-wikidata5M (Chepurova et al., 2023)	0.3180
all-MiniLM	0.3463
all-MPNet	**0.3887**

Table 1: Public test F1 scores. Best checkpoints' scores are reported.

3 Experiments

All fine-tuning experiments have been conducted using LoRA implementation from the `peft` library (Hu et al., 2021). Default LoRA parameters are as follows: lora rank of 16, lora alpha of 32, lora dropout of 0.1. The target modules of LoRA are query and value weight matrices.

Our default model training is conducted for 50 epochs with best checkpoint saving, Binary Cross-entropy loss, batch size of 64, sequence length of 256, AdamW optimizer, learning rate of $3 \cdot 10^{-4}$, default weight decay of 10^{-2}. Additionally, we oversample positive examples proportionally to their disbalance in dataset, which is approximately 10 to 1.

We split the data into train and validation subsets by grouping samples with distinct question in 80:20 proportions correspondingly.

3.1 MiniLM experiments

The MiniLM employed is `all-MiniLM`[1], which is a fine-tuned and diminished version of `MiniLM` by Wang et al. (2020).

The training procedure is default.

3.2 T5 experiments

We fine-tuned `T5-Small`[2] by Chepurova et al. (2023) that was trained on tail and entity prediction in a knowledge graph using the graph's context represented by the node's neighborhood. The result on the public test is presented in Table 1.

The classifier head utilizes the EOS token last hidden representation because of the encoder-decoder architecture. The model was finetuned for 30 epochs with Adafactor optimizer, learning rate of $8 \cdot 10^{-5}$, batch size of 32. Lora alpha was set to 64 for this model.

The input format for this model was corrected to approach the original format the model was trained on. The resulting input format: `'predict [SEP]' + Question + '[SEP]' + Linearized graph + '[SEP]' + Answer Entity`

3.3 MPNet experiments

Another BERT-like model we used is `all-Mpnet-base`[3].

The model was trained for 20 epochs with batch size of 32, sequence length of 200, Adam optimizer, and learning rate of $1 \cdot 10^{-4}$.

4 Additional Experiments

4.1 Ablation study of sequence length and linearized graph usage

The impact of sequence length and linearized graph on performance was examined, see Table 2 . We report F1 score on the public test subset achieved by our best model checkpoints.

SL	Linearized Graph	F1 Score
256	No	0.2276
256	Yes	0.3279
512	**Yes**	**0.3463**

Table 2: Ablation of the Sequence Length (SL) and usage of linearized graph on all-MiniLM performance. Public test scores achieved by best model checkpoints.

4.2 Usage of different backbones

As for phrase-bert, different processing pipeline was utilized according to self-supervised pretraining of this model. In brief, as this model was pretrained for contrastive objective to predict similarity between texts separated by `[SEP]` token. We fine-tuned it with LoRA parameters as described in Section 3, structuring input as `Question entities + ' ' + Question + ' [SEP] + Answer entities`. Information from graph wasn't used during experiments with this model.

5 Conclusion

Naturally, encoder transformer architecture showed the best results on text comprehension tasks. The

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[2] https://huggingface.co/DeepPavlov/t5-wikidata5M-with-neighbors

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

size of the model proved to have a positive influence on its performance once again.

Despite the popularity of T5 model for answer candidates generation, it underperformed in our experiments. Perhaps it is worth utilizing only the encoder part of the model, or using different training procedure.

Another valuable aspect that was confirmed is the avail of graph knowledge for the model. The linearized graph, indeed, provided the model valuable information allowing it to answer the questions better. More advanced subgraph/triplet sampling/generation strategy can avail the model's performance, rendering it a prospective direction of future research.

Limitations

The biggest constraint of our experiments is the reliance on pre-existing subgraphs or triplets derived from the graph. There remains a wide array of potential experiments to be conducted in this area.

Furthermore, we have not investigated the application of in-context learning and prompting techniques with decoder architectures, which could be of even more significant interest due to their current popularity and proven effectiveness.

Acknowledgements

The authors gratefully acknowledge Professor Alexander Panchenko of Skoltech for his guidance during the "Deep Learning for Natural Language Processing" course where this work was conducted.

References

Alla Chepurova, Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. 2023. Better together: Enhancing generative knowledge graph completion with language models and neighborhood information. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5306–5316, Singapore. Association for Computational Linguistics.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Zixian Huang, Ao Wu, Jiaying Zhou, Yu Gu, Yue Zhao, and Gong Cheng. 2022. Clues before answers: Generation-enhanced multiple-choice qa.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiao Li, Gong Cheng, Ziheng Chen, Yawei Sun, and Yuzhong Qu. 2022. Adalogn: Adaptive logic graph network for reasoning-based machine reading comprehension.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering.

Mikhail Salnikov, Hai Le, Prateek Rajput, Irina Nikishina, Pavel Braslavski, Valentin Malykh, and Alexander Panchenko. 2023. Large language models meet knowledge graphs to answer factoid questions.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers.