TIGFORMER at TextGraphs-17 Shared Task: A Late Interaction Method for text and Graph Representations in KBQA Classification Task

Mayank Rakesh[1], **Parikshit Saikia**[2] and **Saket Kumar Shrivastava**[3]
[1]StatusNeo Technology Consulting Pvt Ltd
[2]Greyb Research Pvt Ltd
[3]SLB Pvt Ltd
[1]mayank.rakesh@statusneo.com, [2]parikshit.saikia@greyb.com and [3]Sshrivastava9@slb.com

Abstract

This paper introduces a novel late interaction mechanism for knowledge base question answering (KBQA) systems, combining Graphormer and transformer representations. We conducted extensive experiments, comparing various pooling mechanisms and configurations. Our results demonstrate significant improvements in F1-score compared to traditional baselines. Specifically, we found that attention pooling, in conjunction with linearized graph and question features alongside sub-graph representations, yields the best performance. Our study highlights the importance of advanced interaction mechanisms and the integration of diverse modalities in KBQA systems.

1 Introduction

Transformer models have dramatically reshaped the field of Natural Language Processing (NLP), exemplified by their robust ability to capture intricate textual semantics. Pioneering models such as BERT have revolutionized various NLP tasks, particularly in the domain of question-answering (QA) systems (Devlin et al., 2019). However, despite these advancements, the challenge of accurately answering factoid questions, which often demand precise, context-specific information, continues to pose unique hurdles for language models (Dubey et al., 2019; Sen et al., 2022). While BERT and its successors excel in text processing, their application is sometimes limited without the integration of structured, contextual data from external sources. These challenges often require not just an understanding of the text but also the effective navigation and interpretation of structured knowledge embedded within knowledge graphs. Recent advancements in the field have seen a shift towards leveraging such graphs, which encapsulate rich, interlinked data that can enhance the contextual grounding of answers (Z. Zhang et al., 2020). Knowledge Graphs has been increasingly utilized for solving complex tasks to enrich language models' responses, making them more accurate and contextually aware (Saxena et al., 2020). One notable approach proposes an innovative method of integrating Large Language Models (LLMs) with Knowledge Graphs to enhance QA systems (Salnikov et al., 2023). This method significantly boosts the accuracy of LLMs by using sub-graph extraction based on question entities and re-ranking answer candidates through the linearization of these sub-graphs.

Building upon these foundational insights, this paper aims to further enhance the integration of Language models with Knowledge Graphs. We propose a novel solution that utilizes both transformer-based text embeddings and knowledge graph embeddings more efficiently[1].

2 Related Work

The ability of transformers to capture textual semantics has led towards a lot of research in knowledge and domain adaptation of transformers for question answering tasks (Blom & Pereira, 2023; Cao et al., 2020; Nassiri & Akhloufi, 2023; Yue et al., 2021).

Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, pages 137–141
August 15, 2024

For instance, recent studies have explored the integration and text representations and transformers for KGQA tasks (Lan et al., 2021; Pereira et al., 2022).

The current work is an extension of retrieval-based methods for knowledge base question answering systems. Previous works have analyzed kg-entity-embedding to question embedding comparison and ranking for fetching answer candidates (Razzhigaev et al., 2023; Saxena et al., 2020) or reranking extracted subgraphs using importance prediction (Sun et al., 2019). All these methods are limited to using text features (entities) from these graphs. Thus, treating it as a single modality Natural Language Processing (NLP) problem. (Salnikov et al., 2023) generates candidates using LLMs and uses graph linearization to integrate graph data into t5-transformers for ranking. (Wang et al., 2022) studied the use of convolution neural networks for scoring answer entities in relation to question and graph paths features. (X. Zhang et al., 2022) developed a multi-modal approach fusing text and graph representations. QA-GNN (Yasunaga et al., 2021) explored suggested using message forwarding to update the LM and GNN embeddings simultaneously. Contrasting to all these approaches, the current approach explores the use of transformers and graph transformers representation using multiple late-interaction heads and binary classifying the candidates generated by LLMs using (Salnikov et al., 2023) for correct answer candidates.

3 Methodology

Figure 1 depicts the overview of our late interaction mechanism for Graphormer and transformer representations. The text and graph interactions after pooling are passed through multiple interaction heads. Finally, these fully interacted representations are used to binary classify; question, answer-entity and sub-graph set for correct and incorrect answers.

3.1 Dataset

The dataset[2] (Sakhovskiy et al., 2024) consists of questions with a list of Wikidata entities mentioned in them. For each question, there are 5-10 answer candidates provided, all in the form of Wikidata entities. Additionally, a Wikidata sub-graph is given, which includes the shortest paths between the entities mentioned in the question and the entities listed as answer candidates.

3.2 Text Module

To extract textual information, we embed text representations into a language model encoder. Here we are using a t5-transformer encoder with multi head attention for encoding our representations. The text embedding is represented in Eq. 1.

$$\left\{\widetilde{h_{int}^{(l)}}, \widetilde{h_1^{(l)}}, \cdots, \widetilde{h_T^{(l)}}\right\} = LM\left(\left\{h_{int}^{(l-1)}, h_1^{(l-1)}, \cdots, h_T^{(l-1)}\right\}\right), \tag{1}$$

where $\widetilde{h_{int}^{(l)}}$, denotes representation of text before late interaction.

3.3 Graph Module

We employed Graphormer to depict sub-graph paths and structures (Ying et al., 2021). The native NetworkX encoding of our sub-graphs' structural information includes edge encoding in the attention, spatial (shortest path between node matrices), and centrality (in/out degrees). The graph embeddings are represented with Eq. 2.

$$\left\{\widehat{g_{int}^{(l)}}, \widehat{g_1^{(l)}}, \cdots, \widehat{g_J^{(l)}}\right\} = Graphormer\left(\left\{g_{int}^{(l-1)}, g_1^{(l-1)}, \cdots, g_J^{(l-1)}\right\}\right). \tag{2}$$

Where $\widehat{g_{int}^{(l)}}$ denotes representation of graph before late interaction.

3.4 Interaction Module

Pooled text and graph representations are calculated by passing them through a pooling layer. Mean Pooling, CLS pooling, and attention pooling are used as candidates for our experiments. To interact with and exchange information between text and graph representations, a text-graph interaction module (Eq. 3) is utilized, drawing inspiration from (Lei et al., 2022).

$$\left(g_{int}^{(l)}, h_{int}^{(l)}\right) = inter\left(\widetilde{g_{int}^{(l)}}, \widetilde{h_{int}^{(l)}}\right). \tag{3}$$

The above equation utilizes a inter function. The function first calculates the self and cross modality similarity coefficients between pooled text and graph embeddings:

$$
\begin{aligned}
w_{hh} &= \tilde{h}_{int}^{(l)} \otimes \left(\theta_1 \cdot \tilde{h}_{int}^{(l)}\right), \\
w_{hg} &= \tilde{h}_{int}^{(l)} \otimes \left(\theta_2 \cdot \tilde{g}_{int}^{(l)}\right), \\
w_{gg} &= \tilde{g}_{int}^{(l)} \otimes \left(\theta_2 \cdot \tilde{g}_{int}^{(l)}\right), \\
w_{gh} &= \tilde{g}_{int}^{(l)} \otimes \left(\theta_1 \cdot \tilde{h}_{int}^{(l)}\right),
\end{aligned} \tag{4}
$$

where "⊗" indicates the dot product and θ_1 and θ_2 are hyper-parameters that convert the modality representations into the interaction-sensitive space. The final similarity weights are then obtained using a softmax function:

$$
\begin{aligned}
\widetilde{w}_{hh}, \widetilde{w}_{hg} &= \text{softmax}\left(w_{hh}, w_{hg}\right), \\
\widetilde{w}_{gg}, \widetilde{w}_{gh} &= \text{softmax}\left(w_{gg}, w_{gh}\right).
\end{aligned} \tag{5}
$$

In the end, the interaction modules use the computed similarity weights to enable interaction between the two representations:

$$
\begin{aligned}
h_{int}^{(l)} &= \widetilde{w}_{hh} \tilde{h}_{int}^{(l)} + \widetilde{w}_{hg} \tilde{g}_{int}^{(l)}, \\
g_{int}^{(l)} &= \widetilde{w}_{gg} \tilde{g}_{int}^{(l)} + \widetilde{w}_{gh} \tilde{h}_{int}^{(l)}.
\end{aligned} \tag{6}
$$

3.5 Classification Module

The post interaction graph and text representation are passed through a concatenation layer and finally through a classification head to classify if the set of question, answer entity and sub-graph is factually correct or not.

4 Results and Discussion

The outcomes of the proposed approach are shown in this section, along with comparisons to several baselines. The number of interaction heads for all the experiments were taken as 10 with a learning rate 3e-5. A learning scheduler was using for avoiding overfitting and training was conducted for a total of 10 epochs. F1-score was considered as the evaluation metric. The results are summarized in Table 1. It is demonstrated that, our late interaction framework outperforms the baselines of baseline-text -only, baseline-graph-only and baseline-text-graph. Different pooling mechanisms were compared for the study (Table 1). Attention pooling mechanism outperformed CLS and mean pooling.
Attention Pooling layer with linearized graph and question as text features and sub-graph as graph feature representation inter-action provided best results. We also compared out interaction function with other interaction function such as:

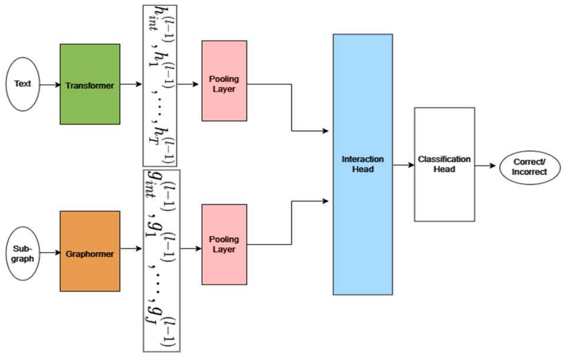

Figure 1: The framework for the proposed interaction method between text and graph representations.

- **Average** Function calculates the average of text and graph representation.
- **Soft** function calculates new representation using two learnable parameters as weights for interaction.

Our proposed method outperforms both interaction mechanism in terms of F1-score.

5 Conclusion

For knowledge base question answering (KBQA) systems, our work presents a novel late interaction method that combines transformer and Graphormer representations. Compared to conventional baselines, this method performs significantly better, showing gains in F1-scores.
We discovered that attention pooling leads to the best classification performance, particularly when combined with question features and linearized graphs with sub-graph representations.
To improve KBQA systems, our research emphasizes the significance of combining various modalities and using cutting-edge interaction methods. By enhancing model robustness and performance through efficient integration of textual and graph-based data, this research paves a path for further improvements in factoid-based question answering.

[1] https://github.com/mayank-rakesh-mck/TextGraphs17-shared-task
[2] https://github.com/uhh-lt/TextGraphs17-shared-task/tree/main

Approach	Pooling Layer	F1-score	Precision	Recall	Accuracy
Baseline-graph-only		0.1266	0.6667	0.0699	0.9103
Baseline-text-graph		0.2022	0.7241	0.1175	0.9138
Baseline-text-only		0.2120	0.1457	0.3888	0.7313
Answer-question-text-subgraph-interaction	Mean Pooling	0.2708	0.2613	0.2811	0.8593
linearized-graph-question-subgraph-interaction	Mean Pooling	0.2996	0.3028	0.2965	0.8711
linearized-graph-question-subgraph-interaction	CLS pooling	0.3054	0.2307	0.4517	0.8090
linearized-graph-question-subgraph-interaction	Attention Pooling	**0.5305**	0.3992	**0.7902**	0.8700

Table 1: A comparison of proposed late interaction method with the baselines

References

Blom, B., & Pereira, J. L. M. (2023). *Domain Adaptation in Transformer Models: Question Answering of Dutch Government Policies* (pp. 196–208). https://doi.org/10.1007/978-3-031-48232-8_19

Cao, Q., Trivedi, H., Balasubramanian, A., & Balasubramanian, N. (2020). *DeFormer: Decomposing Pre-trained Transformers for Faster Question Answering*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Dubey, M., Banerjee, D., Abdelkawi, A., & Lehmann, J. (2019). *LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia* (pp. 69–78). https://doi.org/10.1007/978-3-030-30796-7_5

Lan, Y., He, G., Jiang, J., Jiang, J., Zhao, W. X., & Wen, J.-R. (2021). A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 4483–4491. https://doi.org/10.24963/ijcai.2021/611

Lei, Z., Wan, H., Zhang, W., Feng, S., Chen, Z., Li, J., Zheng, Q., & Luo, M. (2022). *BIC: Twitter Bot Detection with Text-Graph Interaction and Semantic Consistency*.

Nassiri, K., & Akhloufi, M. (2023). Transformer models used for text-based question answering systems. *Applied Intelligence*, *53*(9), 10602–10635. https://doi.org/10.1007/s10489-022-04052-8

Pereira, A., Trifan, A., Lopes, R. P., & Oliveira, J. L. (2022). Systematic review of question answering over knowledge bases. *IET Software*, *16*(1), 1–13. https://doi.org/10.1049/sfw2.12028

Razzhigaev, A., Salnikov, M., Malykh, V., Braslavski, P., & Panchenko, A. (2023). A System for Answering Simple Questions in Multiple Languages. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 524–537. https://doi.org/10.18653/v1/2023.acl-demo.51

Sakhovskiy, A., Salnikov, M., Nikishina, I., Usmanova, A., Kraft, A., Möller, C., Banerjee, D., Huang, J., Jiang, L., Abdullah, R., Yan, X., Ustalov, D., Tutubalina, E., Usbeck, R., & Panchenko, A. (2024, August). TextGraphs 2024 Shared Task on Text-Graph Representations for Knowledge Graph Question Answering. *Proceedings of the TextGraphs-17: Graph-Based Methods for Natural Language Processing*.

Salnikov, M., Le, H., Rajput, P., Nikishina, I., Braslavski, P., Malykh, V., & Panchenko, A. (2023). *Large Language Models Meet Knowledge Graphs to Answer Factoid Questions*.

Saxena, A., Tripathi, A., & Talukdar, P. (2020). Improving Multi-hop Question Answering over Knowledge Graphs using Knowledge Base Embeddings. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4498–4507.

https://doi.org/10.18653/v1/2020.acl-main.412

Sen, P., Aji, A. F., & Saffari, A. (2022). Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering. *Proceedings of the 29th International Conference on Computational Linguistics*, 1604–1619.

Sun, H., Bedrax-Weiss, T., & Cohen, W. (2019). PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2380–2390. https://doi.org/10.18653/v1/D19-1242

Wang, J., Li, W., Guo, Y., & Zhou, X. (2022). Path-aware Multi-hop Question Answering Over Knowledge Graph Embedding. *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, 459–466. https://doi.org/10.1109/ICTAI56018.2022.00074

Yasunaga, M., Ren, H., Bosselut, A., Liang, P., & Leskovec, J. (2021). *QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering*.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., & Liu, T.-Y. (2021). *Do Transformers Really Perform Bad for Graph Representation?*

Yue, Z., Kratzwald, B., & Feuerriegel, S. (2021). Contrastive Domain Adaptation for Question Answering using Limited Text Corpora. *CoRR*, *abs/2108.13854*. https://arxiv.org/abs/2108.13854

Zhang, X., Bosselut, A., Yasunaga, M., Ren, H., Liang, P., Manning, C. D., & Leskovec, J. (2022). *GreaseLM: Graph REASoning Enhanced Language Models for Question Answering*.

Zhang, Z., Liu, X., Zhang, Y., Su, Q., Sun, X., & He, B. (2020). Pretrain-KGE: Learning Knowledge Representation from Pretrained Language Models. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 259–266. https://doi.org/10.18653/v1/2020.findings-emnlp.25