TextGraphs 2024 Shared Task on Text-Graph Representations for Knowledge Graph Question Answering

Andrey Sakhovskiy[1,2 ◇] **Mikhail Salnikov**[2,3 ◇] **Irina Nikishina**[4] **Aida Usmanova**[5]
Angelie Kraft[4] **Cedric Möller**[4] **Debayan Banerjee**[4] **Junbo Huang**[4]
Longquan Jiang[4] **Rana Abdullah**[4] **Xi Yan**[4] **Dmitry Ustalov**[6]
Elena Tutubalina[1,3,7] **Ricardo Usbeck**[4,5] **Alexander Panchenko**[2,3]
[1]Kazan Federal University [2]Skoltech [3]AIRI [4]Universität Hamburg
[5]Leuphana University Lüneburg [6]JetBrains [7]HSE University
{andrei.sakhovskii, m.salnikov}skol.tech firstname.lastname@uni-hamburg.de
aida.usmanova@stud.leuphana.de lastname@airi.net dmitry.ustalov@jetbrains.com

Abstract

This paper describes the results of the Knowledge Graph Question Answering (KGQA) shared task that was co-located with the TextGraphs 2024 workshop.[1] In this task, given a textual question and a list of entities with the corresponding KG subgraphs, the participating system should choose the entity that correctly answers the question. Our competition attracted thirty teams, four of which outperformed our strong ChatGPT-based zero-shot baseline. In this paper, we overview the participating systems and analyze their performance according to a large-scale automatic evaluation. To the best of our knowledge, this is the first competition aimed at the KGQA problem using the interaction between large language models (LLMs) and knowledge graphs.

1 Introduction

Recent years have witnessed remarkable advances in natural language processing (NLP) and network science domains that mostly develop independently with rare intersections. We believe that a proper utilization of graph-based methods for reasoning over a knowledge graph (KG) is a prospective way to overcome critical limitations of the existing large language models (LLMs) which lack interpretability and factual knowledge and are prone to the hallucination problem. In order to encourage novel research efforts that aim to explore the hot topic of LLM prompting from the unique perspective of graph theory, we organized a shared task focused on Knowledge Graph Question Answering (KGQA) as a part of the TextGraphs 2024 workshop on graph-based methods for natural language processing, which was co-located with the ACL 2024 conference hosted on August 15 in Bangkok, Thailand.[2]

The goal of our KGQA shared task was to investigate *how the output of LLMs can be enhanced with KGs*, push the boundaries of current methodologies, and to foster innovative solutions that leverage the strengths of both LLMs and KGs. We formulate the problem as follows. Given an entity from a KG that corresponds to a given textual question, the participating teams have to build a system that classifies whether the entity constitutes the correct answer to this question, or not. The distinctive feature of our task is that it does not only provide pairs of textual question and answer, but provides every pair with a graph representation of the shortest path in KG from entities in the query to the candidate entity generated by an LLM. This setup allows the participants to experiment with different strategies for text and graph information fusion.

The KGQA setup with textual passages anchored to relevant KG subgraphs has been addressed previously. Yasunaga et al. (2022) proposed to pretrain and fine-tune with joint intermodal text-graph interaction on arbitrary text passages linked to ConceptNet (Speer et al., 2017). In LC-QuaD dataset (Trivedi et al., 2017), questions are paired with SPARQL queries for the DBPedia (Lehmann et al., 2015) database. LC-QuaD 2.0 (Dubey et al., 2019) extends LC-QuaD to cover both DBPedia and Wikidata[3] with broader question type coverage.

[1] https://codalab.lisn.upsaclay.fr/competitions/18214

◇ Equal contribution

[2] https://sites.google.com/view/textgraphs2024

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

Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, pages 116–125
August 15, 2024 ©2024 Association for Computational Linguistics

While LC-QuaD's SPARQL queries are inferred from manually curated question-specific SPARQL templates, we stick to the algorithmic approach of Salnikov et al. (2023) to find relevant subgraphs as shortest path KG subgraphs. Thus, we present the first KGQA dataset with a graph construction procedure unified across all questions and Wikidata as reference KG. Previous approaches tried to combine LLMs and KGs by using linearized graphs for fine-tuning (Salnikov et al., 2023; Nikishina et al., 2023) or by fusing encoded representations from a pre-trained Transformer encoder and a graph neural network (Zhang et al., 2022).

The work, as described in this paper, has the following contribution:

- We released a novel dataset for the KGQA binary classification task: given a question, an answer candidate, and a KG subgraph, the goal is to identify whether the provided candidate is a correct answer for the given question using factual information from the graph.
- We organized the open-call shared task and built a public leaderboard to evaluate reasoning-over-graph approaches in a unified controllable set-up by providing questions paired with shortest question-answer paths retrieved from the Wikidata knowledge graph.

Unlike the existing datasets for end-to-end KGQA, our dataset eliminates the potential effect of erroneous entity retrieval, linking, and subgraph retrieval by focusing solely on the fusion phase for textual questions and provided KG subgraph. Thus, it encourages future research focused on cross-modal text and graph interaction.

2 TextGraphs 2024 KGQA Dataset

We constructed a novel dataset for KGQA that was inferred from Mintaka (Sen et al., 2022). Mintaka is a dataset for end-to-end knowledge graph question answering, where each question q is annotated with a set of Wikidata entities $\mathcal{E}_q$ mentioned in the question and ground truth answers A_q for q. Entities from $\mathcal{E}_q$ can serve as anchors for further KG subgraph retrieval and reasoning over the retrieved relevant entities. Although the Mintaka dataset contains the correct answers, we decided to focus on the reasoning part only in our shared task to offer a more controllable environment. In our case, a participating system has to choose the correct answer from a list of possible answer options and the corresponding KG subgraphs.

For our shared task, we followed the KG subgraph construction pipeline proposed by Salnikov et al. (2023). We find the shortest paths between $\mathcal{E}_q$ and the answer candidate entities A_q generated by a language model, such as T5 (Raffel et al., 2020), further linked to the Wikidata KG. The summary of our dataset is presented in Table 1.

Dataset Format. Each instance of the dataset in our shared task was a tuple $s = (q, \mathcal{E}_q, c, \mathcal{G}(\mathcal{E}_q, c), y)$ of the following elements:

- q: question text
- $\mathcal{E}_q$: set of Wikidata entities mentioned in q
- c: candidate answer for q. Unlike Mintaka, we ensure each candidate to be a valid Wikidata entity
- $\mathcal{G}(Q, C)$: a node- and edge-labeled oriented graph obtained as a union of shortest path graphs from each $e \in \mathcal{E}_q$ to candidate c
- y: a binary label describing whether c is a correct answer for q: $y = 1$ if $c \in A_q$ and $y = 0$ otherwise

Data Split. Our dataset was split into two parts:

- The **train set** set consists of 3,535 unique questions inferred solely from Mintaka. We make all ground truth question-candidate binary labels publicly available during the competition.
- The **test set** set covers 1,000 unique questions. 357 of the questions are absent in Mintaka and are manually created and labeled with ground truth answer entities from scratch. The test set consists of two subparts: (i) public and (ii) private with 700 and 300. The private part (300 unique questions) includes newly created questions exclusively.

2.1 Wikidata Knowledge Graph

Wikidata is a collaborative knowledge graph that contains nearly two billion facts, covering a diverse range of topics including geography, history, famous people, and events.[4] It serves as a centralized

[4] https://www.wikidata.org/wiki/Wikidata:Statistics

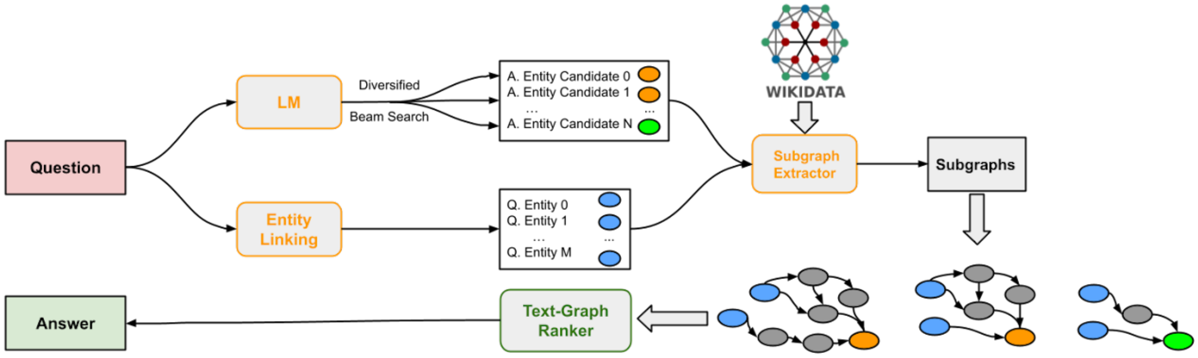

Figure 1: Overview of the pipeline for TextGraphs 2024 shared task workflow. We link entities mentioned in question and answer candidates generated by an LM to Wikidata KG. Then, we extract shortest path subgraphs between question and candidates. Question and candidate graph can be further passed to a ranking network to obtain a confidence score of a candidate being the correct answer to the given question.

repository for structured data and supports various Wikimedia projects and external applications. The data in Wikidata can be accessed through a public SPARQL endpoint.[5] However, due to the large volume of information, the endpoint is limited to shorter queries. Nevertheless, Wikidata is fully downloadable, allowing users to locate all the data on local servers and bypass public endpoint restrictions by using SPARQL query engines or graph databases such as iGraph, which we have used to manage our local Wikidata dump.

2.2 Answer Candidate Generation

To generate an initial set of answers, we use the T5-large language model (Raffel et al., 2020), which has been trained on the Mintaka training dataset (Sen et al., 2022). To increase the diversity of the generated responses, we employ Diverse Beam Search (Vijayakumar et al., 2016), a generalized framework for producing a list of varied sequences, which may be used instead of the traditional beam search approach.

Partition	# Questions	# Candidates
Train	3,535	36,762
Test (Total)	1,000	10,961
Public Test	700	7,694
Private Test	300	3,267

Table 1: Summary of the dataset used in the shared task.

2.3 Candidate Entity Linking

Entity linking with Wikidata involves identifying and linking entities to their corresponding entries in the Wikidata knowledge graph using generated strings. This process can be challenging due to the large number of entities in Wikidata, variations in their names, and the high ambiguity of entity mentions. To address these challenges, modern neural network-based approaches require extensive processing (Cao et al., 2022). For our shared task, we used the public Wikimedia APIs[6] that use search engines to retrieve entities based on their labels and aliases. By indexing Wikidata entities and their associated textual data in a search engine like Elasticsearch, which is used by the public Wikimedia API, we can efficiently retrieve candidate entities through queries based on their profiles generated from contextual mentions.

2.4 Answer Candidate Filtering

While our answer candidate generation and linking pipeline could produce an arbitrary number of negative samples, we aimed at mining a small subset of only the hardest ones. We assumed that a harder negative candidate entity should be semantically similar to the ground truth answer. For example, for a question “In Greek mythology, who stole fire from Olympian gods to give it to humanity?” with the correct answer “Prometheus” (Titan, culture hero, and trickster figure in Greek mythology), a more challenging negative sample would be “Hermes” (Olympian god in Greek religion and

[5] https://query.wikidata.org

[6] https://www.wikidata.org/w/api.php

mythology) rather than "Pythia" (priestess of the Temple of Apollo at Delphi). For a given question q and ground truth answers set A_q, we sampled a random true answer $a \in A_q$. Next, we ranked each negative candidate c with respect to the semantic similarity of its description $\mathrm{desc}(c)$ to the description $\mathrm{desc}(a)$ of a. As a similarity measure, we adopted the mutual implication score[7] (MIS), a RoBERTa$_{large}$-based (Liu et al., 2019) similarity metric designed for paraphrase detection (Babakov et al., 2022). For each question, we truncated its negative candidate count to 9 having the highest MIS score and removed questions with less than five negative candidates.

2.5 Subgraph Construction

We associated each question-answer pair with the corresponding induced subgraph from the Wikidata KG. This subgraph was generated by extracting the shortest paths between an entity derived from the question and a candidate answer entity, and then by identifying all distinct nodes along these paths. The extraction process also preserves all edges between these nodes, ensuring that relationships between the entities in the question and answer are maintained. The goal of this approach was to create a comprehensive representation of relevant information from the KG for each question-answer pair, accurately reflecting the connections between entities present in the original graph. Figure 2 shows simple examples of the obtained shortest path graphs.

3 Shared Task Description

Typically, an end-to-end KGQA pipeline includes multiple subtasks, such as named entity recognition and entity linking of entities mentioned in a question; construction of a KG subgraph for reasoning. It is challenging in multi-step KGQA pipelines to determine whether a prediction error comes from inaccurate entity retrieval and linking, or the model failed to perform reasoning over a fine-grained and informative graph. In our shared task, we used a simplified setup with fixed question-candidate subgraphs to enable more accessible evaluation of knowledge graph reasoning systems.

3.1 Baselines

As baselines, we adopted three supervised approaches built upon a BERT-based encoder (Devlin et al., 2019) and ChatGPT[8] model as a zero-shot LLM-based baseline. Additionally, we reported the quality for constant baseline. For non-LLM baselines, the task was formulated as a binary classification task: each question-candidate pair is labeled with either 1 or 0 independently of other candidates.

For three encoder-based supervised baselines, we adopt encoder-only MPNet[9] model (Song et al., 2020) as a base model and perform a 9:1 train/validation split. Each model is trained for five epochs with a batch size of 64 using Adam optimizer (Kingma and Ba, 2015) and cross-entropy loss. For prediction, we load each model's parameters from the best epoch in terms of validation F$_1$ score. The classification threshold of 0.5 remains constant for all three baselines.

Graph Linearization. For shortest path graphs representation, we adopt the graph linearization format from Salnikov et al. (2023) to represent each candidate graph as a textual string. We traverse graph edges starting from question entities $\mathcal{E}_q$ moving to candidate answer entities $\mathcal{E}_c$. Each labeled edge (h, r, t) of type r starting in h leading to t is linearized as "h, r, t". If either $h = c$ or $t = c$, they were additionally emphasized with BERT model's `[SEP]` tokens: e.g., "`[SEP]` h `[SEP]` r `[SEP]` t" if $h = c$. A linearized graph $\mathcal{L}(\mathcal{G}(\mathcal{E}_q, c))$ for question q and candidate c was obtained as a concatenation of all its linearized edges.

Text-Only Baseline. This baseline completely ignored the presence of question-candidate graphs and learned to classify textual question-candidate pairs. Specifically, we pass concatenated question and candidate string "q `[SEP]` c" to a binary classifier, where `[SEP]` was a special separator token of a BERT-based model.

Graph-Only Baseline. This baseline aimed to explore what quality a model would demonstrate seeing only linearized graph $\mathcal{L}(\mathcal{G}(\mathcal{E}_q, c))$ without even knowing what question q produced graph $\mathcal{G}(\mathcal{E}_q, c)$.

Text+Graph Baseline. As a simple joint text-and-graph reasoning baseline, we adopted a binary classifier over the concatenation of question and linearized candidate graph following (Salnikov et al., 2023): "q `[SEP]` $\mathcal{L}(\mathcal{G}(\mathcal{E}_q, c))$".

[7] https://huggingface.co/s-nlp/Mutual_Implication_Score

[8] https://chat.openai.com

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

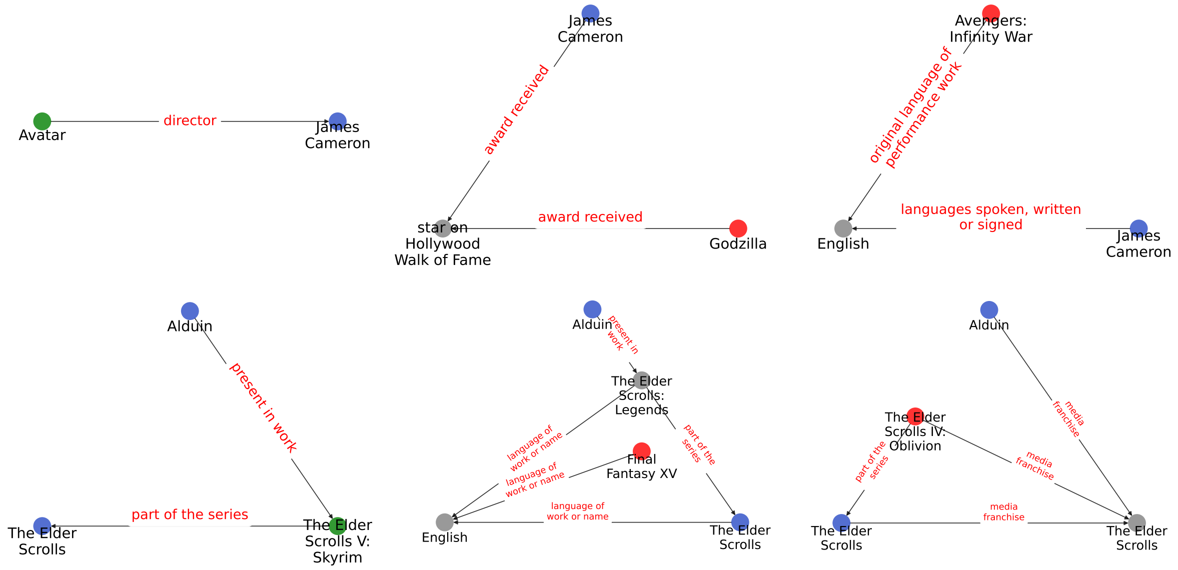

Figure 2: Question-candidate graph visualizations. **First row:** question "Which film directed by James Cameron became the highest-grossing movie of all time?" and three answer candidates: (i) Titanic (true answer), (ii) Godzilla, and (iii) Avengers: Infinity War. **Second row:** question "Which game is in The Elder Scrolls series and has Alduin as the main villain?" Entities mentioned in question (e.g., "James Cameron") are colored blue, intermediate nodes on the path from question entities to a candidate are colored grey. Correct and incorrect answer candidates are colored in green and red, respectively

ChatGPT Baseline. As an LLM baseline, we adopted ChatGPT version *gpt-4-0613*. To let the model differentiate between candidate answers with matching textual names but different underlying Wikidata entities, we modified ambiguous answer choices by adding the type of graph edge leading to the candidate node in the question-candidate graph. For instance, for the question "Which film directed by James Cameron became the highest-grossing movie of all time?" there are two candidate answers named "Titanic": (i) 1997 film by James Cameron and (ii) 1953 film by Jean Negulesco. The types of edges leading to these two candidate answers are "director" and "original language of performance work". Table 2 shows an example of the prompt.

Constant Baseline. This baseline assigned label 1 to all samples, i.e., marked all candidate answers as being correct.

3.2 Evaluation

Our shared task was deployed on the Codalab competition platform.[10] All submitted systems were evaluated on precision, recall, and F_1-score for positive class as well as classification accuracy. We performed the ranking of submitted systems based on F_1 score. Overall, the task consisted of three phases: development, evaluation, and post-evaluation.

Development Phase. This phase started with the release of the labeled train set on March 10, 2024. The participants were invited to get acquainted with the data format and to start their preliminary experiments. The phase can be considered closed with the release of the test set on March 25, 2024.

Evaluation Phase. On March 25, 2024, we released the test set with no ground truth labels provided. The set consisted of both public and private subsets, but the participants were not informed of what subset each test sample belongs to. At this stage, the participants were encouraged to submit test set prediction to the **public** leader board which provided evaluation results for the public test subset. By the end of the evaluation phase on May 6, 2024, participants were allowed to make their final submission to obtain evaluation scores on both **private** and **public** subsets. Private evaluation results were made publicly available after May 6, 2024.

Post-Evaluation Phase. After the end of Shared Task's official evaluation part on May 6, 2024, all participants can make submissions on the test data.

[10] https://codalab.lisn.upsaclay.fr/competitions/18214

Baseline	Input Examples
Text-Only	- Which film directed by James Cameron became the highest-grossing movie of all time? </s> Avatar - Which film directed by James Cameron became the highest-grossing movie of all time? </s> Titanic
Graph-Only	- </s> Titanic </s>, director, James Cameron - </s> Godzilla </s>, award received, star on Hollywood Walk of Fame James Cameron, award received, star on Hollywood Walk of Fame
Text+Graph	- Which film directed by James Cameron became the highest-grossing movie of all time? </s> </s> Titanic </s>, director, James Cameron - Which film directed by James Cameron became the highest-grossing movie of all time? </s> </s> Godzilla </s>, award received, star on Hollywood Walk of Fame James Cameron, award received, star on Hollywood Walk of Fame
ChatGPT	**Please answer the following question.** **provide one or more comma-separated option ids as an answer.** Which film directed by James Cameron became the highest-grossing movie of all time? 0. Avatar 1. Avengers: Infinity War 2. Godzilla 3. Home Alone 4. Home Alone: The Holiday Heist 5. Spectasia 6. Terminator 2: Judgment Day 7. Terminator II 8. The Terminator 9. Titanic (director) 10. Titanic (original language of performance work)

Table 2: Input examples for baseline models; </s> is a separator token of the MPNet encoder used for baselines.

These submissions have a separate leaderboard and are not considered for the official public evaluation summary.

4 Official Results

In total, we have received submissions from 30 teams, including both public and private leaderboards. After the end of the evaluation phase, we asked the participants to describe their systems.

4.1 Shared Task Submissions

Team NLPeople applied the Chain-of-Thought (CoT, Wei et al. (2022)) technique to decompose the target question into a series of sub-questions and attempted to use question-specific prompts based on question types (Moses et al., 2024). The final prediction is an ensemble of three LLM-based solutions: (1) Llama3-70B-Instruct[11] with CoT, (2) GPT-3.5 with CoT, and (3) Llama3-70b-instruct with Question-Specific prompts. In cases when the ensemble failed to make a prediction, a zero shot GPT-4's prediction was reported.

Team HW-TSC implemented an LLM prompt design based on self-ranking and emotional incentives (Tang et al., 2024). Self-ranking implied that the `gpt-4-1015-preview` base model is asked to score its answer choices with confidence levels. Emotional prompts were aimed at encouraging the model to carefully examine a question.

Team Skoltech adopted question-candidate graph sizes and Wikidata entity description as additional features to enhance the initial GPT-4 pre-

[11] `https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct`

	Private Evaluation					Public Evaluation				
Team Name	**Rank**	**P**	**R**	**F$_1$**	**Acc**	**Rank**	**P**	**R**	**F$_1$**	**Acc**
NLPeople	1	**86.67**	85.14	**85.90**	**97.39**	1	**86.54**	85.45	**86.00**	**97.41**
HW-TSC	2	84.34	82.11	83.21	96.91	2	83.95	81.96	82.94	96.87
Skoltech	3	81.78	84.26	83.00	96.78	3	81.05	84.34	82.66	96.71
POSTECH	4	82.50	80.65	81.56	96.60	4	82.14	80.42	81.27	96.56
Baseline: ChatGPT	5	59.91	78.59	67.99	93.09	5	58.11	78.18	66.67	92.73
Team <blank>	6	60.54	72.73	66.07	93.03	6	58.20	71.47	64.16	92.58
BpHigh	7	55.99	75.86	64.43	92.18	—	—	—	—	—
tigformer	8	40.39	80.35	53.76	87.10	7	39.93	79.02	53.05	87.00
CUFE	9	51.92	55.52	53.66	91.05	—	—	—	—	—
Team_87	10	65.13	42.72	51.59	92.52	8	63.71	43.22	51.50	92.44
Iron Autobots	11	73.15	35.68	47.96	92.77	9	77.05	32.87	46.08	92.85
nlp_enjoyers	12	40.90	39.98	40.43	89.01	10	39.29	38.46	38.87	88.76
NLPunks	13	40.61	37.83	39.17	89.03	12	41.23	32.87	36.58	89.41
KseniiaPetrushina	14	34.90	44.18	39.00	87.10	11	33.96	40.28	36.85	87.17
Transformers-Spring24	15	29.19	44.48	35.24	84.75	16	28.56	40.70	33.56	85.03
Cordyceps	16	35.60	34.80	35.20	88.04	15	34.43	33.71	34.06	87.87
YAR	17	40.99	30.69	35.10	89.41	17	42.70	26.99	33.08	89.85
Transformers-Spring24	18	34.27	34.70	34.48	87.69	—	—	—	—	—
Fancy Transformers	19	39.31	30.01	34.04	89.14	19	38.64	27.83	32.36	89.19
xren	20	48.40	22.19	30.43	90.53	22	51.16	21.68	30.45	90.80
ThangDLU	21	18.56	67.55	29.12	69.31	20	22.29	58.04	32.21	77.29
adugeen	22	50.14	17.89	26.37	90.68	18	46.63	25.17	32.70	90.37
Baseline: Text+Graph	23	64.88	15.35	24.82	91.32	28	72.41	11.75	20.22	91.38
IRRRR	24	15.27	57.77	24.16	66.14	25	16.03	59.16	25.22	67.40
YAR	25	65.61	14.17	23.31	91.30	—	—	—	—	—
Baseline: Text-Only	26	15.04	38.32	21.60	74.04	27	14.57	38.88	21.20	73.13
mathamateur (–)	27	10.01	86.51	17.95	26.17	21	35.49	26.85	30.57	88.67
Baseline: Constant	28	9.33	**100.00**	17.07	09.33	29	9.29	**100.00**	17.01	9.2
Baseline: Graph-Only	29	62.16	6.74	12.17	90.91	30	66.67	06.99	12.66	91.03
hawkeoni	30	23.86	2.05	3.78	90.25	13	24.68	68.11	36.24	77.72
Hijli_JU_NLP	31	17.65	1.47	2.71	90.17	31	22.64	1.68	3.12	90.33
a063mg	—	—	—	—	—	14	42.71	28.67	34.31	89.80
russabiswas	—	—	—	—	—	23	28.47	29.23	28.85	86.60
__Team1()__	—	—	—	—	—	24	43.37	18.74	26.17	90.17
GrahamSquad	—	—	—	—	—	26	70.78	15.24	25.09	91.54

Table 3: Official evaluation results of the TextGraphs 2024 Shared Task for the public and private evaluation phases. **P**, **R**, **F$_1$**, and **Acc** stand for positive class precision, recall, F$_1$-score, and accuracy, respectively. The best values for each metric are highlighted in **bold**. Official baselines are highlighted in cyan.

dictions (Lysyuk and Braslavski, 2024). They rephrased the given questions to further question rephrasing technique, we further strengthen their prediction.

Team <Blank> used `gpt-3.5-turbo` enhanced with CoT and XML tags prompting techniques.

Team tigformer implemented a late interaction for exchanging information between text and graph representations via an attention-pooling layer (Rakesh et al., 2024). They employed Graphformer (Ying et al., 2021) to encode question-candidate graphs and a T5 model (Raffel et al., 2020) to obtain textual representations.

Team nlp_enjoyers fine-tuned an MPNet encoder (Song et al., 2020) using LoRA (Hu et al., 2022). For a given question q and candidate c, they modified the input format for Text+Graph baseline as: "$\mathcal{E}_q$: q [SEP] $\mathcal{L}(\mathcal{G}(\mathcal{E}_q, c))$" and separated each edge in the linearized graph with a semicolon (Kurdiukov et al., 2024). They assumed only a single candidate for each question to be correct and reformulated the task from binary classification to

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

ranking: given a question, they select the most probable answer based on model scores for all candidate answers.

Team Fancy Transformers experimented with different graph characteristics including length, density, degree centrality, eigenvector centrality, closeness centrality and PageRank. They adopted the encoder-only all-MiniLM-L12-v2[12] and all-MiniLM-L6-v2[13] models for encoding textual information. They reported that the latter model achieved higher performance.

Team JellyBell applied Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) approach to answering questions (Belikova et al., 2024). They retrieved relevant to question documents from internet by DuckDuckGo API[14] and generated answer by prompting LLM with fetched documnets.

5 Discussion

Table 3 presents the evaluation results for both public and private phases of the TextGraphs 2024 shared task on knowledge graph question answering. Three teams have managed to outperform a strong ChatGPT baseline with their LLM-based systems showing that large models are good at memorizing factual knowledge during pre-training.

Since one of the primary goals of our shared task was to evaluate the ability of LMs to reason over given KG subgraphs, we highlight the teams that submitted non-LLM solutions. **Team tigformer** has gained rank 8 of 31 with 53.76% F_1-score using two separate encoders for textual and graph information with late intermodal interaction. It is worth noting that while **Team nlp_enjoyers** only achieved the 12th place on the private leaderboard, they managed to surpass the initial Text+Graph baseline by 15.6% F_1-score with light-weighted modifications only. Their results indicate that while resource-demanding and computationally expensive LLM dominate the task in general, there might be some room for improving light-weighted task-specific solutions. **Team Fancy Transformers** has achieved 34.04% F_1-score using *all-MiniLM-L6-v2* encoder having 22.7M parameters only enhanced with classical non-neural graph features.

6 Conclusion

We presented an overview of the TextGraphs 2024 shared task on knowledge graph questions answering (KGQA) with pre-calculated shortest path graphs for Wikidata entities mentioned in the question and a candidate answer. Analysis of the results has revealed that large language models (LLMs) currently show superior performance even in a very simplified binary classification task formulation when a model is asked to find the right answer among the pre-defined set of answer candidates. While LLMs are extremely resource-demanding, the exploration of effective light-weighted systems for question-oriented graph representation and reasoning still remains a challenge

We hope that our competition will encourage further research on developing effective reasoning methods over retrieved KG subgraphs, exploring novel subgraph representation techniques, and improving the interpretability and explainability of the resulting question answering models.

Acknowledgements

Participants affiliated with Universität Hamburg extend their gratitude to NFDI4DS (National Research Data Infrastructure for Data Science) for their gracious support and endorsement of this shared task. We acknowledge the computational resources of HPC facilities at the HSE University. We would like to thank all the participating teams who contributed to the success of the shared task through their interesting approaches and experiments.

References

Nikolay Babakov, David Dale, Varvara Logacheva, and Alexander Panchenko. 2022. A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 300–321, Dublin, Ireland. Association for Computational Linguistics.

Julia Belikova, Evegeniy Beliakin, and Vasily Konovalov. 2024. Jellybell at textgraphs-17 shared task: Fusing large language models with external knowledge for enhanced question answering. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke

[12] https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2

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

[14] https://duckduckgo.com

Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Trans. Assoc. Comput. Linguistics*, 10:274–290.

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.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 69–78. Springer.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Nikita Kurdiukov, Pavel Tikhomirov, Viktoriia Zinkovich, and Sergey Karpukhin. 2024. nlp_enjoyers at textgraphs-17 shared task: Text-graph representations for knowledge graph question answering using all-mpnet. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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. *CoRR*, abs/1907.11692.

Maria Lysyuk and Pavel Braslavski. 2024. Skoltech at textgraphs-17 shared task: Finding gpt-4 prompting strategies for multiple choice questions. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Movina Moses, Vishnudev Kuruvanthodi, Mohab Elkaref, Shinnosuke Tanaka, James Barry, Geeth De Mel, and Campbell D Watson. 2024. Nlpeople at textgraphs-17 shared task: Chain of thought questioning to elicit decompositional reasoning. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Irina Nikishina, Polina Chernomorchenko, Anastasiia Demidova, Alexander Panchenko, and Chris Biemann. 2023. Predicting terms in IS-a relations with pre-trained transformers. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 134–148, Nusa Dua, Bali. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Mayank Rakesh, Parikshit Saikia, and Saket Kumar Shrivastava. 2024. Tigformer at textgraphs-17 shared task: A late interaction method for text and graph representations in kbqa classification task. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

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. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 635–644, Hong Kong, China. Association for Computational Linguistics.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Wei Tang, Xiaosong Qiao, Xiaofeng Zhao, Min Zhang, Chang Su, Yuang Li, Yinglu Li, Yilun Liu, Feiyu Yao, Shimin Tao, Hao Yang, and He Xianghui. 2024. Hw-tsc at textgraphs-17 shared task: Enhancing inference capabilities of llms with knowledge graphs. In *Proceedings of the TextGraphs-17: Graph-based Methods for Natural Language Processing*, Bangkok, Thailand. Association for Computational Linguistics.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 210–218. Springer.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28877–28888.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D. Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.