Prompt engineering for tail prediction in domain-specific knowledge graph completion tasks

Davide Mario Ricardo Bara[1]

[1]*Babes-Bolyai University, Business Informatics Research Center, Cluj-Napoca, Romania*

Abstract

In this paper we propose a system architecture for tackling the LM-KBC Challenge 2024 [1] task as a tail prediction problem in a knowledge graph completion setup. We experiment with several prompting techniques as suggested in the TELER taxonomy [2]. We notice that under the few-shot paradigm, using In-Context-Learning by supplying the LLM with valuable public information about the query subject and expert knowledge about the relation under study, we can improve the tail prediction performance.

1. Introduction

Knowledge bases (KBs) are structured repositories that store data in a way that is easily retrievable and useful for various applications. These systems emulate human understanding by organizing information into a structured format such that machines can process them efficiently. Knowledge graphs (KG) store the facts of a knowledge base as triples (s, p, o), where s is the subject of the relationship, p represents the predicate or the relation and o represents the object. Thus, as the subject or the object of each triple could be any entity in the target domain, the KG is in fact a huge graph allowing one to navigate through it. We can distinguish between (i) general and publicly available KGs, like DBPedia, Yago or Wikidata encompassing general knowledge available in the world or (ii) KGs storing specific data related to some particular domain of interest, either from the scientific or the business environment.

KGs are seen as a more expressive way of representing data than the traditional databases, as they allow one to run semantic queries, enabling advanced capabilities of information discovery, eventually through reasoning. But, in many situations, such KBs and their associated KGs store incomplete information, or particular KGs are improperly aligned with one another, although storing the same logical information. Thus, various tasks like triple classification, head prediction, tail prediction, relation prediction or entity classification are of interest and they are generally known as knowledge graph completion tasks.

Recent research [3, 4] showed that Large Language Models can be employed to solve such tasks with increasingly better performance. LLMs are deep learning (DL) models comprising billion of parameters learned on large amounts of publicly available data, trying to model the generative likelihood of word sequences in order to correctly predict the next token that should come in a sentence. Generally, they show good performance in responding to questions related

KBC-LM '24: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2024

dmrbara@gmail.com (D. M. R. Bara)

to their training data - thus general knowledge questions, but they are limited and known to hallucinate when responding questions on specific data outside their training.

In the proposed challenge we face a tail prediction problem on a KB containing specific data. Thus, the designed system should respond to the following question: given a specific knowledge base supplied as a training dataset, how can we enhance the tail entity prediction accuracy, leveraging a LLM of a limited capacity? This research question is challenging by itself as tail identification could be solved as a reasoning problem and LLMs' ability for logical reasoning is unclear [5]. Moreover, LLMs are known to hallucinate [6] when responding questions outside the coverage of their training data.

In [7] the authors have shown that prompt engineering has a massive influence over the capacity of various LLMs to generate triples out of a free text, therefore, in our system we employ multiple prompting techniques, as suggested in [2], transmit to the LLM the relevant information from the training data and the real world and combine the LLM response with publicly available knowledge in order to enhance its performance over the tail prediction task.

The structure of the paper is as follows: Section 2 reviews related work in this field. Section 3 describes our research methodology, including the dataset and validation strategy, the processing steps followed by our system and the prompts used to interogate the LLM. Section 4 presents the results obtained by running the system with various prompts and the expected system performance over unseen data. The last section concludes the paper.

2. Related work

In this section, we aim to provide an overview of existing research efforts related to knowledge base construction tasks and the application of LLMs in this domain.

Language models have significantly transformed research in the field of natural language processing (NLP) in recent years. It has evolved from optimizing predictors and architecture to the paradigm of pre-training and model optimization, and recently to the paradigm where a pre-trained model is queried and provides a prediction [8]

Due to the large datasets used, pre-trained language models contain various types of knowledge stored in their parameters, thus becoming possible alternatives to traditional knowledge bases. The latter do not offer the same flexibility in terms of information eloquence and expansion [8]. At the same time, creating and maintaining knowledge bases is a complex process that involves a series of challenges [9]. An important aspect is the effort required to create a knowledge base, which involves extracting relational data from large unstructured texts through complex NLP pipelines. Among the advantages of language models are the fact that they do not require the creation of a validation schema and their ability to handle queries from multiple domains [10].

The use of language models as knowledge bases has been done using a series of strategies: integrating knowledge at the time of pre-training, connecting an external base to a pre-trained model, using an attention mechanism, and using a method based on extracting necessary nodes from a knowledge graph. [8]

A report on the applicability of language models in specific knowledge base tasks [8] presents as a major impediment the interpretation and updating of knowledge stored in the model's parameters. Unlike traditional knowledge bases, the information in a pre-trained language model is stored in a loosely structured manner, making it difficult to control.

The authors of the same report believe that future research should focus on three important aspects: improving the interpretability of language models, increasing their ability to store and extract specific information, and ensuring consistency of responses regardless of the questions asked or the language used.

3. Methodology

In this section we present the methodology of our study. The source code of the system and the results over the test dataset are available on Github at https://github.com/davidebara/lm-kbc. Experiments were performed on a Pay-as-you-go Google Colab subscription using a L4 GPU with Python 3.10.12.

Our final architecture uses Meta's Llama 3 8B Instruct[1] model, the best ranked LLM in LMSYS' ChatbotArena that has less than 10B parameters.

3.1. Datasets and validation strategy

Challenge organizers supplied the labeled data (triples with known object entities) in two files, in total making available 755 unique subject entities linked via 5 relations to a bigger number of object entities [1]. The train dataset contains 377 subject entities, while the validation dataset contains 378 subject entities. To design a validation strategy for being able to assess the confidence in our results, we merged these 2 datasets and created 50 train / validation datasets using stratified random sampling, by keeping the same number of subject entities in each dataset and preserving the given distribution of subject entities per relation. All results reported below are average values computed on the validation datasets over 50 experiments. The final submitted system makes use of the full training data available.

As the dataset contains a very limited number of subject entities and the available computing capabilities were very limited forbidding us to perform a fine-tuning experiment of a LLM with enough capacity, we decided to use prompting techniques together with exploiting publicly available information to enhance the performance of the tail prediction task. With this respect, we approached Wikidata[2], a free and open knowledge base who can be seen as a huge knowledge graph and can be interrogated in a programmatic way via SPARQL queries. We performed a manual and implicit relation alignment task identifying the relations in the Wikidata which are relevant from the semantic point of view with the relations supplied in the training data.

3.2. System architecture

In this subsection we present the architecture of our system, detailing the processing steps performed to produce the requested output.

Algorithm of fig. 1 presents the steps pursued in order to process a test dataset. The algorithm iterates over all $(subject, relation)$ pairs of the dataset and for each pair it formats a

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct and https://ollama.com/library/llama3:instruct

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

```
function PROCESS-SUBJECT-RELATION-PAIRS(dataset) : returns results
    $results \leftharpoonup \emptyset$
    for (subject, relation) in dataset do
        $context \leftharpoonup$ FETCH-KG-CONTEXT(subject, relation)
        $prompt \leftharpoonup$ GENERATE-PROMPT(subject, context, relation)
        $prediction \leftharpoonup$ CALL-LLM(prompt)
        $answers \leftharpoonup$ DISAMBIGUATE(prediction)
        $entities \leftharpoonup$ LINK-TO-REAL-WORLD(answers)
        $results \leftharpoonup results \cup entities$
    end for
    return $results$
end function
```

Figure 1: General algorithm for processing all items of a test dataset

prompt according to one of the prompt levels of the TELER taxonomy [2] and sends a query to the selected LLM with the formatted prompt. Next, predictions supplied by the LLM are disambiguated and aligned to real-world knowledge extracted from Wikidata. In the last step, obtained entities are returned as a response.

When generating the prompt in the GENERATE-PROMPT function we applied specific prompt templates for each of the 5 relations supplied in the training dataset. Prompt templates are found in the $prompt\_templates$ folder. The LLM receives clear instructions about the task it needs to solve and some relevant context that could be publicly identified about the subject of the query. Prompts are described in subsection 3.3.

Function FETCH-KG-CONTEXT interrogates Wikidata to eventually discover publicly available information about the subject. SPARQL query templates are found in the `relations` folder. For each relation in the training dataset, we developed a specific query, aiming to incorporate as much expert knowledge as possible. Responses supplied by the SPAQRL query are transmitted as context for the LLM query. In this function, we perform a logical entity alignment task for a knowledge graph, with Wikidata as the target.

We ask the LLM to supply textual information in a given format. For prompt templates that require output examples, we provide the LLM with two question-response pairs, to indicate how the textual output should be formatted for both existing and non-existing object entities. This is essential for the levels 5 and 6 of prompting, the ones that prove to work better than all the previous ones.

DISAMBIGUATE function extracts the entities provided by the LLM from its response, by parsing the JSON output and removing all unnecesary details. LINK-TO-REAL-WORLD function checks the Wikidata knowledge base to determine if the LLM responded with information available there. If a match is found, it returns the corresponding entity identifier.

3.3. Prompt engineering

In this section we present the prompt templates formatted according to the TELER taxonomy [2]. Since the prompts were integrated with the research framework provided by the competition, the LLM might receive additional text based on the system's setup.

3.3.1. Level 0 prompts

Level 0 prompts only provide the LLM with raw data. In our case, this means the model will only receive the name of the subject entity. As you could notice from section 4, the performance for this prompt is worse than that of the baseline model. No context information is supplied, either from the training dataset or from the real world.

3.3.2. Level 1 prompts

Level 1 prompts provide a basic one-sentence directive that states a high-level goal for the LLM. Our results show that this approach performs slightly better than Level 0 prompting, achieving results comparable to the baseline model provided in the competition.

```
Provide the names of the stock exchanges that list {subject_entity} shares.
```

Figure 2: The level 1 prompt template for the "companyTradesAtStockExchange" relation.

3.3.3. Level 2 prompts

Level 2 prompts include sub-tasks that need to be executed by the LLM in order to achieve the high-level goal. The results indicate that incorporating a chain-of-thought approach can improve LLM performance by guiding the model through a clear, logical sequence of steps and reducing the ambiguity of the initial goal.

```
Provide a comma-separated list of all stock exchanges where shares of
{subject_entity} are traded. Include only the names, and if none, state {None}.
Perform the task in distinct steps: extract relevant financial data about
{subject_entity}, verify the stock exchanges where its shares are traded from
reliable sources, and cross-check the information for accuracy. Ensure your response
is clear, accurate, and concise.
```

Figure 3: The level 2 prompt template for the "companyTradesAtStockExchange" relation.

3.3.4. Level 3 prompts

Level 3 prompts extend the level 2 prompts by providing the LLM with a structured list of sub-tasks rather than a paragraph of instructions. Up to this level, the prompts fulfill the Zero-shot prompting paradigm, as presented in [11].

```
Provide a comma-separated list of all stock exchanges where shares of
{subject_entity} are traded. Include only the names, and if none, state “None”.
Perform the task in distinct steps: 1. Extract relevant financial data about
{subject_entity}. 2. Verify the stock exchanges where its shares are traded from
reliable sources. 3. Cross-check the information for accuracy. 4. Ensure your
response is clear, accurate, and concise.
```

Figure 4: The level 3 prompt template for the "companyTradesAtStockExchange" relation.

3.3.5. Level 4 prompts

Level 4 prompts build upon level 3 prompts, adding guidelines on how the answer will be evaluated or expected answer examples.

Here, we move towards a few-shot prompting paradigm, more precisely to *In-Context-Learning*, by adding relevant examples of a solution to the given task. For each relation type we provided two examples of how the LLM should respond when prompted about a specific subject entity, one where a right answer exists and one where it doesn't.

```
Provide a comma-separated list of all stock exchanges where shares of {subject_entity} are traded. Include only the names, and if none, state “None”. Perform the task in distinct steps: 1. Extract relevant financial data about {subject_entity}. 2. Verify the stock exchanges where its shares are traded from reliable sources. 3. Cross-check the information for accuracy. 4. Ensure your response is clear, accurate, and concise. Follow the format of the provided examples: Example 1: “Provide a comma-separated list of all stock exchanges where shares of Apple Inc. are traded. Response: NASDAQ, Frankfurt Stock Exchange, Swiss Exchange.” Example 2: “Provide a comma-separated list of all stock exchanges where shares of a private company are traded. Response: None.”
```

Figure 5: The level 4 prompt template for the “companyTradesAtStockExchange” relation.

3.3.6. Level 5 prompts

Level 5 prompts include information supplied in level 4 prompting, with the addition of context gathered via retrieval-based techniques from Wikidata. As shown in the results section, the predictive performance of the LLM improves significantly.

To retrieve relevant context from Wikidata, we created five SPARQL queries, each tailored to a specific relation type. Regardless of whether entities are found, the results are appended to the end of the prompt template within the script that runs the model. The template sentence is structured as follows: “`Query results for subject “{subject_entity}” and property “{relation_type}” on the Wikidata Knowledge Graph: {query_result}`”. If no information is returned, `query_result` will be the string ”None”.

```
Provide a comma-separated list of all stock exchanges where shares of {subject_entity} are traded. Include only the names, and if none, state “None”. Perform the task in distinct steps: 1. Extract relevant financial data about {subject_entity}. 2. Verify the stock exchanges where its shares are traded from reliable sources. 3. Cross-check the information for accuracy. 4. Ensure your response is clear, accurate, and concise. Follow the format of the provided examples: Example 1: “Provide a comma-separated list of all stock exchanges where shares of Apple Inc. are traded. Response: NASDAQ, Frankfurt Stock Exchange, Swiss Exchange.” Example 2: “Provide a comma-separated list of all stock exchanges where shares of a private company are traded. Response: None.” Utilize additional information fetched through reliable information retrieval techniques to confirm the accuracy of the stock exchanges list.
```

Figure 6: The level 5 prompt template for the “companyTradesAtStockExchange” relation.

3.3.7. Level 6 prompts

Level 6 includes all elements of the level 5 prompts, with an explicit statement asking the LLM to explain its own output. From the obtained results, we notice that the performance of the LLM does not improve.

```
Provide a comma-separated list of all stock exchanges where shares of
{subject_entity} are traded. Include only the names, and if none, state “None”.
Perform the task in distinct steps: 1. Extract relevant financial data about
{subject_entity}. 2. Verify the stock exchanges where its shares are traded from
reliable sources. 3. Cross-check the information for accuracy. 4. Ensure your
response is clear, accurate, and concise. Follow the format of the provided examples:
Example 1: “Provide a comma-separated list of all stock exchanges where shares of
Apple Inc. are traded. Response: NASDAQ, Frankfurt Stock Exchange, Swiss Exchange.”
Example 2: “Provide a comma-separated list of all stock exchanges where shares of
a private company are traded. Response: None.” Utilize additional information
fetched through reliable information retrieval techniques to confirm the accuracy
of the stock exchanges list. Justify your response with a detailed explanation of
the sources and reasoning process. If stating ’None’, explain why the information
is unavailable or not applicable.
```

Figure 7: The level 6 prompt template for the “companyTradesAtStockExchange” relation.

3.3.8. Final prompt template

As level 5 prompts prove to be the best ones, we selected them in the final solution. Figure 8 illustrates the selected prompt template. Additionally, it assigns a specific role to the LLM for each task and clear instructions on how the response should be formatted to avoid extraneous tokens in the answers. We also removed any extraneous tokens like numbers or bullets of lists, as the text is sent to the LLM as a paragraph either way.

```
You are a financial expert. Provide a comma-separated list of all stock exchanges
where shares of {subject_entity} are traded. Include only the names, and if none,
state “None”. Perform the task in distinct steps: extract relevant financial data
about {subject_entity}, verify the stock exchanges where its shares are traded from
reliable sources and cross-check the information for accuracy. Ensure your response
is clear, accurate, and concise. Follow the format of the provided examples: Example
1: “Provide a comma-separated list of all stock exchanges where shares of Apple
Inc. are traded. Response: NASDAQ, Frankfurt Stock Exchange, Swiss Exchange.”
Example 2: “Provide a comma-separated list of all stock exchanges where shares
of a private company are traded. Response: None.” Utilize additional information
fetched through reliable information retrieval techniques to confirm the accuracy
of the stock exchanges list.
```

Figure 8: The final prompt template for the “companyTradesAtStockExchange” relation.

Table 1
Results from running the system with various prompting techniques.

Prompt level	Relation	Macro-F1	Micro-F1	Avg. number of preds.	Empty preds.
Level 0	awardWonBy	0.029	0.030	10.000	1
	companyTradesAtStockExchange	0.475	0.488	0.890	18
	countryLandBordersCountry	0.666	0.719	3.015	4
	personHasCityOfDeath	0.320	0.202	0.540	49
	seriesHasNumberOfEpisodes	0.040	0.054	0.480	57
	All relations	0.341	0.183	1.312	129
Level 1	awardWonBy	0.068	0.041	9.800	0
	companyTradesAtStockExchange	0.481	0.466	1.140	22
	countryLandBordersCountry	0.875	0.888	2.500	14
	personHasCityOfDeath	0.480	0.316	0.400	61
	seriesHasNumberOfEpisodes	0.150	0.182	0.650	36
	All relations	0.453	0.220	1.288	133
Level 2	awardWonBy	0.056	0.048	10.700	2
	companyTradesAtStockExchange	0.559	0.484	1.110	42
	countryLandBordersCountry	0.950	0.928	2.441	17
	personHasCityOfDeath	0.450	0.320	0.450	55
	seriesHasNumberOfEpisodes	0.130	0.173	0.500	51
	All relations	0.474	0.230	1.267	167
Level 3	awardWonBy	0.068	0.054	11.400	0
	companyTradesAtStockExchange	0.532	0.522	1.010	18
	countryLandBordersCountry	0.908	0.906	2.368	17
	personHasCityOfDeath	0.420	0.321	0.570	43
	seriesHasNumberOfEpisodes	0.117	0.144	0.670	34
	All relations	0.448	0.229	1.323	112
Level 4	awardWonBy	0.053	0.049	10.700	2
	companyTradesAtStockExchange	0.544	0.447	1.090	42
	countryLandBordersCountry	0.927	0.917	2.500	18
	personHasCityOfDeath	0.360	0.293	0.610	39
	seriesHasNumberOfEpisodes	0.100	0.119	0.680	34
	All relations	0.434	0.223	1.362	135
Level 5	awardWonBy	0.276	0.135	13.300	0
	companyTradesAtStockExchange	0.997	0.994	0.800	35
	countryLandBordersCountry	0.914	0.945	2.721	19
	personHasCityOfDeath	0.937	0.939	0.600	41
	seriesHasNumberOfEpisodes	0.950	0.960	0.980	2
	All relations	0.934	0.416	1.471	97
Level 6	awardWonBy	0.273	0.133	13.200	0
	companyTradesAtStockExchange	0.981	0.969	0.820	36
	countryLandBordersCountry	0.916	0.947	2.676	19
	personHasCityOfDeath	0.910	0.900	0.650	38
	seriesHasNumberOfEpisodes	0.910	0.875	1.080	1
	All relations	0.913	0.407	1.505	94

4. Results

Table 1 presents the results of running the architecture presented in subsection 3.2 with various prompt styles supplied to the Llama 3 8B Instruct model.

We notice that prompts formatted at levels 1 to 4 give results in the margin of $0.43 - 0.47$

for the macro-F1 score, which are similar to the baseline results supplied by the LLM. This is expected, as the LLM does not get any additional information from the real world or from the specific topic under discussion. While we add context to the prompt, we notice that the object retrieval performance get higher than 0.9 for the Macro-F1 score. Here we notice that only in half of the cases, the LLM produces responses exactly from the context, thus we anticipate that the LLM brings in added value in providing the response.

With the randomization scheme presented in the methodology, we are able to compute the confidence interval for the macro-F1 score. Fig. 4 presents the Macro-F1 scores derived from the 50 experiments. As fig. 4 shows, most experiments achieved a Macro-F1 score between 0.91 and 0.94. There were also two experiments whose Macro-F1 scores were below 0.91, highlighting the issue of inconsistency in the responses provided by the language models, which is in-line with existing literature[12].

The 95% confidence interval ranges from 0.927 to 0.932, with an average Macro-F1 score of 0.929. This indicates that the model performs well and consistently, despite variations in the training datasets.

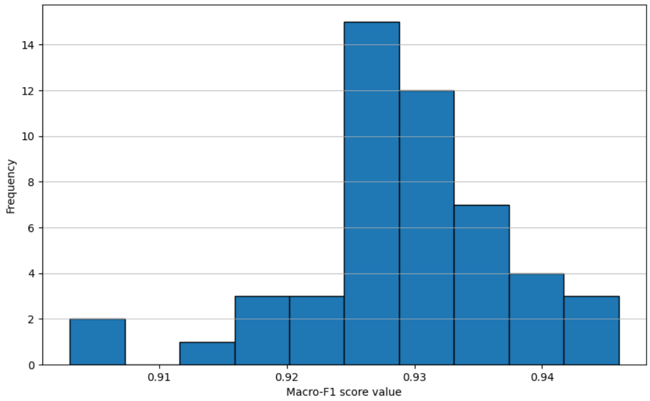

Figure 9: Histogram of the obtained Macro-F1 scores

5. Conclusion

As observed in our experiments, the ability of a pre-trained language model to accurately predict information about a specific entity is limited by the dataset it was trained on. Gradually optimizing the prompts proved to be a crucial step in improving an LLM's predictive performance. Although there were a few exceptions, our model achieved a good confidence interval, suggesting low variation between the responses provided.

Future research should investigate the entity alignment task between two KGs - which could drastically improve prediction performance, how an LLM chooses between multiple conflicting sources of information, the reasoning process behind its answers and ways to control the generated answers.

References

[1] J.-C. Kalo, S. Razniewski, T.-P. Nguyen, B. Zhang, Knowledge base construction from pre-trained language models 2022, in: Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models, CEUR-WS, 2024.

[2] S. K. K. Santu, D. Feng, TELeR: A general taxonomy of LLM prompts for benchmarking complex tasks, in: H. Bouamor et al. (Ed.), Findings of the ACL: EMNLP 2023, Singapore, 2023, ACL, 2023, pp. 14197–14203. doi:10.18653/V1/2023.FINDINGS-EMNLP.946.

[3] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying Large Language Models and knowledge graphs: A roadmap, CoRR abs/2306.08302 (2023). doi:10.48550/ARXIV.2306.08302.

[4] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, LLMs for knowledge graph construction and reasoning: Recent capabilities and future opportunities, CoRR abs/2305.13168 (2023). doi:10.48550/ARXIV.2305.13168.

[5] J. Huang, K. C. Chang, Towards reasoning in large language models: A survey, in: A. Rogers, J. L. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023, Association for Computational Linguistics, 2023, pp. 1049–1065. doi:10.18653/V1/2023.FINDINGS-ACL.67.

[6] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, S. Shi, Siren's song in the AI ocean: A survey on hallucination in large language models, CoRR abs/2309.01219 (2023). doi:10.48550/ARXIV.2309.01219.

[7] V. I. Iga, G. C. Silaghi, Assessing llms suitability for knowledge graph completion, CoRR abs/2405.17249 (2024). doi:10.48550/ARXIV.2405.17249.

[8] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, M. Ghazvininejad, A review on language models as knowledge bases, CoRR abs/2204.06031 (2022). doi:10.48550/ARXIV.2204.06031.

[9] G. Weikum, X. Dong, S. Razniewski, F. Suchanek, Machine knowledge: Creation and curation of comprehensive knowledge bases, Foundations and Trends in Databases 10 (2021) 108–490. doi:10.1561/1900000064.

[10] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, CoRR abs/1909.01066 (2019). doi:10.48550/arXiv.1909.01066.

[11] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. Wen, A survey of large language models, CoRR abs/2303.18223 (2023). doi:10.48550/ARXIV.2303.18223.

[12] Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. H. Hovy, H. Schütze, Y. Goldberg, Measuring and improving consistency in pretrained language models, Trans. Assoc. Comput. Linguistics 9 (2021) 1012–1031. doi:10.1162/TACL_A_00410.

A. Online Resources

The source files of our system can be accessed through GitHub.