Knowledge-centric Prompt Composition for Knowledge Base Construction from Pre-trained Language Models

Li, X.; Hughes, A.; Llugiqi, M.; Polat, F.; Groth, P.; Ekaputra, F.J.

Publication date
2023

Document Version
Final published version

Published in
Joint proceedings of the 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC)

Knowledge-centric Prompt Composition for Knowledge Base Construction from Pre-trained Language Models

Xue Li[1], Anthony Hughes[2,5], Majlinda Llugiqi[3], Fina Polat[1], Paul Groth[1] and Fajar J. Ekaputra[3,4]

[1]University of Amsterdam, Amsterdam, The Netherlands
[2]University of Wolverhampton, Wolverhampton, UK
[3]Vienna University of Economics and Business, Vienna, Austria
[4]TU Wien, Vienna, Austria
[5]Data Language, London, UK

Abstract

Pretrained language models (PLMs), exemplified by the GPT family of models, have exhibited remarkable proficiency across a spectrum of natural language processing tasks and have displayed potential for extracting knowledge from within the model itself. While numerous endeavors have delved into this capability through probing or prompting methodologies, the potential for constructing comprehensive knowledge bases from PLMs remains relatively uncharted. The Knowledge Base Construction from Pre-trained Language Model Challenge (LM-KBC) [1] looks to bridge this gap. This paper presents the system implementation from team *thames* to Track 2 of LM-KBC. Our methodology achieves 67 % F1 score on the test set provided by the organisers outperforming the baseline by over 40 points, which ranked 2nd place for Track 2. It does so through the use of additional prompt context derived from both training data and the constraints and descriptions of the relations. All code and results can be found on GitHub[1].

1. Introduction

The field of Artificial Intelligence (AI) has seen huge improvements in tasks related to language due to Pre-trained Language Models (PLMs)[2] and the computational efficiency introduced by transformers [3]. This significant improvement can be seen in areas such as translation, summarisation, and classification [4, 5, 6].

Given their effectiveness in many information extraction tasks [5, 7], there has been a movement by the community to study their use in tasks focused specifically on knowledge base construction [8, 9]. As part of that larger interest, The Knowledge Base Construction from Pre-trained Language Model Challenge (LM-KBC) was launched in 2022 to better understand

[1]Code and results available - https://github.com/effyli/lm-kbc/

KBC-LM'23: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2023

x.li3@uva.nl (X. Li); a.j.hughes2@wlv.ac.uk (A. Hughes); majlinda.llugiqi@wu.ac.at (M. Llugiqi); f.yilmazpolat@uva.nl (F. Polat); p.t.groth@uva.nl (P. Groth); fajar.ekaputra@wu.ac.at (F. J. Ekaputra)

CEUR Workshop Proceedings (CEUR-WS.org)

the role that PLMs can play as a source of knowledge themselves [10]. Essentially, providing a framework to study how one can construct a knowledge base directly from a PLM.

This report presents our approach and results for the second edition of the LM-KBC challenge at the 22nd International Semantic Web Conference (ISWC 2023). The challenge is to predict objects for a given subject-predicate pair. An example is given a subject, *Matt Damon*, and the relationship we are targeting, *person has number of children*, retrieve the object, in this case, a number, for that pair. It may be that the model needs to predict an existing Wikidata object, example being, the subject country *Fiji*, has associated geographical states, and the objects we wish to retrieve are those Wikidata states.

We propose a pipeline for knowledge base construction by prompting large language models, specifically, GPT-3.5 and GPT-4. We explore different setups with in-context learning by utilizing an example selector and knowledge-enriched prompts to provide more contextually relevant prompts. Our results show rule-based example selectors considering cardinality per relation exhibit significant performance on the task. Furthermore, enriching entities and relations with additional properties obtained from GPT-4 help boost the performance even further.

2. Related Work

The notion of using a language model as a source of knowledge itself was brought to the fore by the LAMA paper in 2019 [11]. This can be seen as one part of the larger move towards prompting PLMs to solve NLP tasks. We refer the reader to the survey by Min et al. [12] for a deeper dive into prompting and associated architectures for NLP. Here, we focus on work directly related to the LM-KBC challenge. An overview of the various approaches can be found in the 2022 challenge introduction [10].

Specifically, our work follows on from the winner of task 2 of last year's challenge, "Prompting as Probing"[13]. In their work, they prompt GPT3 with manually curated prompt templates, including 4 examples from the training set in their prompts. These are then updated with the specific subject entity of interest during the prompting workflow. Additionally, they include a post-processing step called "fact-probing" in which the PLM is asked to judge whether a given result produced by PLM is indeed true. This helps improve the precision of the model. The authors went on to perform an ablation study outside of the challenge whereby they utilised Wikidata to help improve entity disambiguation during the post-processing step. By using Wikidata information, such as the hypothesised concepts type in relation to the relationship used during prediction, to validate the prediction. This study proved an slight performance gains, however this was not allowed to be part of the reporting in 2022, but in 2023, such retrieval augmentation is allowed. We employ a similar approach here.

Our approach differs because we focus on dynamically selecting examples from the training set to include in the prompt. Additionally, our prompts provide more context than those used by Alivanistos et al [13]. Also, we note that we use a newer version of GPT.

In [14], a benchmark is provided to establish the ability of models to construct knowledge graphs from text. The authors provide an ontology description as part of their prompts. The prompt consistently employs the *relation(subject, object)* format to represent relationships and expects the model's output to adhere to this notation. We also employ ontology descriptions

(i.e. extra knowledge base context) in our prompts.

3. LM-KBC Challenge Definition

The Language Model Knowledge Base Construction (LM-KBC) challenge task is defined as follows. Take a set of subject (s) predicate (p) pairs ($< s, p >$) and predicting a set of objects ($o_1, o_2...$) in relation to those pairs. The target set of objects can be; (1) a wikidata identifier, (2) a numerical value, or (3) empty.

The LM-KBC Challenge provides two distinct tracks for participants. The first, known as the "Small-model Track," restricts participants from using pre-trained Language Models with no more than 1 billion parameters and excludes the use of contextual information. The second termed the "Open Track," imposes no limitations on the model size and permits the inclusion of contextual data. For the purposes of our research, we tackle the Open Track.

3.1. Dataset

The dataset available for LM-KBC comprises 5820 samples (i.e. triples) evenly divided over training, validation and test set. The objects within the test set were withheld during the time period when our system was developed. The dataset encompasses an array of 21 distinct relations, where a diverse range of subject-entities are provided for each relation. Each triple contains both the Wikidata identifiers and also lexicalizations of each element of the triple as English text.

Each relation in the train and validation sets is accompanied by a set of ground truth object-entities, curated to align with specific subject-relation pairs. It is noteworthy that the length of object-entities affiliated with a given subject-relation pairing exhibits variability. Meaning, in the test set, the implementation must correctly predict sets of objects and empty sets.

We note that there are four relations, e.g. *PersonHasPlaceOfDeath*, where there are potentially zero relations to the subjects in the available sets. In comparison, *CountryHasStates* requires from between one and twenty objects for the prediction of the related subjects. Furthermore, objects are not limited to other Wikidata entries, the entries could also be numerical, e.g. *SeriesHasNumberOfEpisodes*.

4. Methodology

In-context learning is a fundamental capability in many language modelling approaches. The approach is prominent in GPT models particularly starting from GPT-3 [15]. Our approach centers around in-context learning (i.e. prompting, few-shot learning), which combines the capabilities of pre-trained language models with the contextual information available in the text [16]. Specifically, we focus on the design and utilization of few-shot prompts. Prompting refers to the use of specific instructions and/or statements to induce the model to complete certain tasks. Few-shot learning is an approach where the language model learns how to perform a task from minimal data points, i.e., learning the task from only a few examples (few shots). Considering prompting and few-shot learning principles, we carefully designed our prompts that

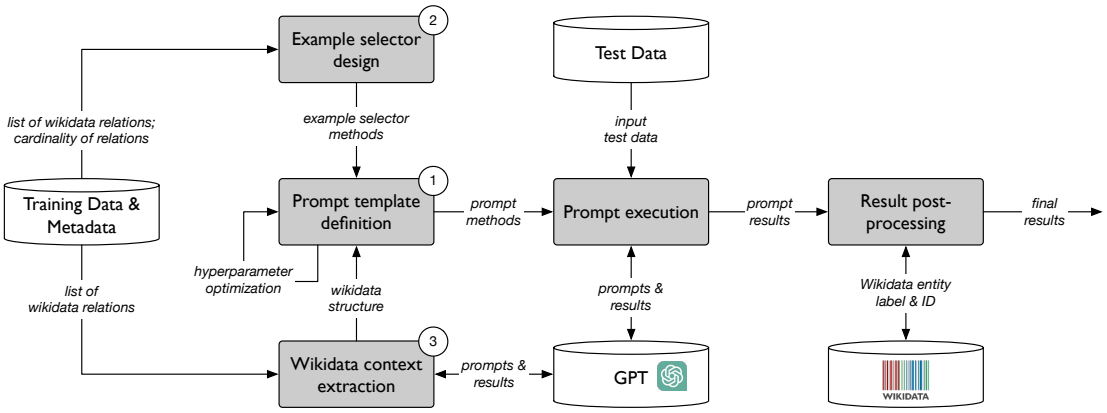

Figure 1: An overview of the *thames*' team method.

integrate both static and dynamic elements. An overview of our method is provided in Figure 1. From the training dataset, we derive a list of wikidata relations, along with their cardinality distribution. We use the set of relations for two purposes: example selector design and Wikidata context extraction that is prompted from GPT models. The prompt template is subsequently defined by these two components, followed by a refining process for hyperparameters such as the number of examples selected based on the final model performance on the evaluation set. We then execute the prompts with test data through GPT models. Finally, the generations are post-processed and connected to Wikidata IDs. We will explain the components of our approach in the following sub-sections.

4.1. Prompt Template Definition

In compiling our prompts, we employ static prefix and suffix components while dynamically selecting examples in between based on the given subject-predicate pair. We incorporate static elements at the beginning and end of each prompt to provide a consistent context for the language model. The prefix scopes the prompt, guiding the model to understand the relevant parameter space, while the suffix ensures structure and uniformity. The crux of our methodology lies in selecting and integrating dynamic examples from the training dataset into few-shot prompts. This process is facilitated by the use of two distinct example selectors, designed to guide the language model's comprehension of the extraction task at hand.

GPT-3.5 and GPT-4 have been fine-tuned utilizing dialogue and instruction datasets [17]. On top of this finetuning, they are both optimized for dialogue and instruction followed by using Reinforcement Learning with Human Feedback (RLHF) [6]. RLHF is a machine-learning approach that involves mapping out optimal strategies based on human responses. This technique allows the language model to learn more complex behaviours and concepts that are difficult to define or specify explicitly in a traditional reinforcement learning setup. By incorporating humans in training, the model can inherit a more nuanced understanding of several tasks [18]. Leveraging this background knowledge, we carefully phrase the static components of the prompts.

Initially, the prefix assigns a role to the LLM, i.e. "Act as a knowledge base", "Imagine that you are Wikidata", etc. Then, a brief task description is given, followed by an explicit statement indicating that the prompt will continue with examples. A fixed example template has been devised to be populated by the example selectors. The suffix then delineates the conclusion of the selected examples, stating that it is now the LLM's turn for prediction. The prompt ends with a template to be filled with the input subject-predicate pair and a signal of continuation, i.e. `" : "`, `" [ "`, etc.

4.2. Example Selectors for In-Context Learning

Context sensitivity is a recognized phenomenon in in-context learning [15]. The immediate textual content that appears prior to the prediction point is the sole form of the input. Everything the model generates from that point is a continuation of the prompted input. This sensitivity can be both beneficial and problematic. An advantage is that it enables the model to adapt rapidly to changing task requirements and examples. As a drawback, the high sensitivity can lead to issues with model consistency and predictability, resulting in a hallucinatory generation. Therefore, prompts play a crucial role when it comes to extracting knowledge from the language model.

We account for context sensitivity in the selection of both static and dynamic components of the prompt. However, the dynamic selection of the most relevant examples is specifically designed to leverage context sensitivity. Our example selectors pick out the most relevant instances from the training set. The rule-based selector follows certain rules for the selection while the similarity-based selector leverages cosine similarity. The selectors are detailed in the following subsections.

4.2.1. Rule-based Example Selector

The rule-based example selector is designed as a systematic approach to sample examples from the training set. Given that instances may have zero or more objects, this approach ensures that the diverse nature of examples is taken into account. To instil this understanding, we enriched our prompts with five specific examples for each instance. The selection criteria for these five examples are as follows:

- Minimum Object Example: We selected one example with the fewest number of objects for a given relation.
- Maximum Object Example: We selected one example with the highest number of objects for a given relation.
- Random Selection: To add an element of variability and ensure broader coverage, we incorporated three additional examples. These were chosen at random from the training set.

This strategy helps in achieving a balanced representation of the data, ensuring that the model does not develop a bias towards any particular pattern.

4.2.2. Similarity-based Example Selector

The similarity-based example selector operates by using semantic similarity measures to identify instances that are akin to the input instance. This approach allows for the dynamic selection of examples that are contextually compatible with the input text. The functioning of this system relies on embeddings and necessitates a list of vectorized or embedded examples, to which the given input can be compared. Furthermore, it computes a semantic similarity score, such as cosine similarity or dot product, in order to select the closest examples from the pool of embedded examples.

As far as performance is concerned when applied to GPT-3.5, this selector is noticeably slower than the rule-based selector, which is expected given its operation at the embedding level. Semantic similarity-based selection methods are more suited to tasks that harbour a high degree of variation and ambiguity. However, the task at hand in this case, shows a lower degree of variation as it is limited to 21 relations.

4.3. Prompt Improvement through Wikidata Context Extraction

We hypothesise, that given a subject-predicate pair, we can gain greater accuracy when predicting the object if the model is given the correct context for that pair. We extend this further by utilising the schema and knowledge base from which the subject and predicate came from. Specifically for this task, we state that, for a subject-predicate pair from Wikidata, it is possible to use the qualities from that particular knowledge base to enhance the prompt. To this end, we prompt GPT-3.5 to provide a set of relevant contexts related to the given properties. The prompt that we use to extract these contexts is available in our GitHub repository[1]. An excerpt of the result is available in Listing 1

Listing 1: An excerpt of the extracted Wikidata context on the given properties from GPT-3.5

```
{
  "CompanyHasParentOrganisation": {
    "value": "P749",
    "wikidata_id": "P749",
    "wikidata_label": "parent organization",
    "domain": "organization",
    "range": "organization",
    "explanation": "This property is used to indicate the parent organization
        of a company."
  },
  ...
}
```

An example context usage is provided in Listing 2. For this example, the subject, *AT&T*, the Wikidata ID *Q35476* and relation, *CompanyHasParentOrganisation*, are provided by the competition dataset. Using Wikidata context generated through prompt, we are able to enhance the context by finding related information to the given relation. These contexts include subject and object class type, domain and range information, the label of the given relation, and a full

[1] https://bit.ly/3QDewyx

description of that label. This context information is injected into the relevant sections of the prompt.

Listing 2: An example wikidata context usage within the prompt

```
Your task is to predict objects based on the given subject and relation.
- Given Subject: ('AT&T','Q35476')
- Subject Type: 'organization'
- Object Type: 'organization'
- Relation: 'CompanyHasParentOrganisation'
- Relation Wikidata ID: 'P749'
- Relation Label (Wikidata): 'parent organization'
- Relation Explanation (Wikidata): 'This property is used to indicate
    the parent organization of a company.'
==>
Predicted Objects:
```

4.4. Prompt Execution and Post-processing

We adapted prompt execution and post-processing parts of the baseline code provided by the challenge. The adaptations are mostly to cater for the needs of debugging and testing. One exception, however, pertains to the post-processing entity with title and subtitle in the results (i.e., results containing character `":"`). We noticed that some results in the validation set only matched results without the subtitle part. Therefore, we add a slight modification in the Wikidata disambiguation to check for only the main title in case of full string did not return Wikidata IDs. This update helps to improve the results, especially for the `PersonHasAutobiography` relation. Additionally, with our post-processor, we notice that model tends to generate duplicated results and therefore we added a de-duplication step.

5. Results

We summarise the results of our system implementation and experiments in Table 1. We then present a more detailed comparison between GPT-3.5 and GPT-4 with the highest performing example selection methodology utilised when prompting each model, this comparison is shown in Table 2. Also discussed are the results for zero-object cases, this is where the system should correctly *not* predict an object for some subject-predicate pairs.

5.1. Overview

In our methodology, we discuss two potential example selection mechanisms, where the selected examples are injected into various prompts. We first performed experiments with GPT-3.5 and the similarity-based selection methodology, we then used our proposed rule-based methodology for both models. From our experiments, we find that a rule-based approach to prompt creation yields greater scores in recall and F1 regardless of the underlying model. The use of GPT-4 in combination with the rule-based approach gave the best results overall for all F1 metrics.

Table 1
This table presents the results for each of the presented prompt selection methodologies, and for each model utilized in the experiments. The highlighted block presents the highest score per metric.

Model	Selector	Precision	Recall	F1
GPT-3.5	Similarity-based	0.5595	0.6154	0.5484
	Rule-based	0.6105	0.6492	0.5863
GPT-4	Rule-based	**0.7128**	**0.6894**	**0.6744**

5.2. Rule-based prompts

Given that rule-based prompts offer the best results overall in our experiments, we present a more detailed comparison between GPT-3.5 and GPT-4 where a full breakdown of all predicate scores is available in Table 2.

Table 2
This table presents a side-by-side comparision between GPT-3.5 and GPT-4. Each relation has a breakdown of its precision, recall and F1 against a respective model. The highlighted block presents the highest score per metric.

	GPT-3.5			GPT-4		
Relation	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
BandHasMember	0.4998	0.6186	0.5110	0.5507	0.6408	**0.5628**
CityLocatedAtRiver	0.7200	0.5393	0.5885	0.7700	0.5882	**0.6375**
CompanyHasParentOrganisation	0.3400	0.7350	0.3367	0.6400	0.7900	**0.6367**
CompoundHasParts	0.9073	0.8960	0.8983	0.9667	0.9710	**0.9677**
CountryBordersCountry	0.8815	0.7783	**0.8156**	0.8905	0.7543	0.7898
CountryHasOfficialLanguage	0.6364	0.8756	0.6548	0.9218	0.8244	**0.8474**
CountryHasStates	0.7353	0.7156	0.7124	0.7384	0.7436	**0.7381**
FootballerPlaysPosition	0.5600	0.7383	0.6173	0.6850	0.7167	**0.6897**
PersonCauseOfDeath	0.6400	0.7400	0.6400	0.8000	0.8033	**0.7950**
PersonHasAutobiography	0.3081	0.3150	0.3008	0.3517	0.4350	**0.3742**
PersonHasEmployer	0.4860	0.3265	**0.3675**	0.4600	0.3020	0.3422
PersonHasNoblePrize	0.9500	0.9500	0.9500	0.9900	0.9900	**0.9900**
PersonHasNumberOfChildren	0.5000	0.5000	0.5000	0.6500	0.6500	**0.6500**
PersonHasPlaceOfDeath	0.5000	0.6700	0.5000	0.6700	0.7400	**0.6733**
PersonHasProfession	0.3650	0.3312	0.3230	0.5150	0.4086	**0.4268**
PersonHasSpouse	0.6983	0.7050	**0.6983**	0.7033	0.7150	**0.6983**
PersonPlaysInstrument	0.5903	0.4661	0.4801	0.7700	0.4752	**0.5540**
PersonSpeaksLanguage	0.7622	0.8385	0.7062	0.8485	0.8427	**0.7964**
RiverBasinsCountry	0.7253	0.8279	0.7249	0.8434	0.9336	**0.8542**
SeriesHasNumberOfEpisodes	0.4450	0.5100	0.4667	0.5750	0.5800	**0.5767**
StateBordersState	0.5696	0.5568	0.5197	0.6278	0.5724	**0.5616**
Average	0.6105	0.6492	0.5863	**0.7128**	**0.6894**	**0.6744**

Table 2 demonstrates the efficacy of GPT-4 over GPT-3.5. GPT-4 outperforms in 18 of

the 21 relations that require predictions. The two relations where GPT-3.5 outperforms are *CountryBordersCountry* and *PersonHasEmployer*, while for one relation *PersonHasSpouse*, the two models are tied. We look to further break down the relations in GPT-4 to identify any patterns that may emerge. The three lowest performing classes are *PersonHasAutobiography*, *PersonHasEmployer*, *PersonHasProfession*. All three of these relations are the subject of type Person. This pattern of poorer performance on Person related relations is common to both models.

5.3. Zero-object cases

As discussed in Section 3.1, it is possible for one of the given subject-predicate pairs that the target object to be predicted is empty. In Table 5.3, we present the F1 scores in regard to this specific issue.

Table 3
This table presents the results of zero-object detection for only the rule-based selection methodology across the two models utilized in the experiments. The highlighted block represents the highest score per metric.

Model	Precision	Recall	F1
GPT-3.5	0.6348	0.6854	0.6591
GPT-4	**0.7037**	**0.8920**	**0.7867**

Overall, GPT-4 is the better-performing model for this specific task when using rule-based example selection for prompt construction.

6. Discussion

The findings and observations from our study have shed light on a few perspectives when using PLMs for KBC, including how contexts affect in-context-learning performance, the impact of post-processing and what the limitations of GPT-family models are for this task.

Contextual Relevance in In-Context Learning: In our experiments, we observe that both demonstrated examples and additional knowledge of the entities play a crucial role in enhancing a model's understanding and generation. This aligns with the fundamental idea that a richer context helps produce a more coherent response. To improve contextual relevance, one can select more relevant demonstrations given relations and entities to predict. Additionally, providing extra knowledge for relations and entities can help generate more accurate responses.

Impact of Post-Processing: PLMs do not always follow the given instructions. Due to the fact that PLMs are often fine-tuned with natural question-answering style tasks, the generation of answers often comes as a natural language-style answer. Hence, being able to unify the answers and quantitatively evaluate them is a challenge. Follow from this is that effective post-processing strategies are necessary for the generation of quality results.

Performance Enhancement of GPT-4: Our results corroborate the general consensus that GPT-4 improves performance compared to its predecessors, such as GPT-3.5.

Hallucinations on Relation Types: Although GPT-4 has shown significant ability for predicting the objects given subject and relation, the model still shows signs of hallucination. The model especially struggles with specific types of relations such as *PersonHasProfession* and *PersonHasEmployer*. When allowed to generate multiple answers, the model tends to hallucinate after the first correct answer and generate related professions but not factually correct. This might be improved if the model can fact-check with every answer it produces.

Temporal misalignment between Wikidata and GPT-family Models: The dataset from the organizers is from Wikidata, which contains up-to-date knowledge, while GPT-family models were only trained with text till September 2021, resulting in a performance bottleneck due to the nature of the dataset.

7. Conclusion & Future work

We proposed a PLM-based pipeline centered on in-context learning for performing knowledge base construction, specifically, for the task of predicting objects given a subject and relation. We explored different approaches to prompting including the use of contextual information from training and an associated knowledge graph. Our results indicate that providing examples with higher contextual relevance, including the type of relations, and the possible cardinality of the objects, can help with knowledge base construction.

Our results show that PLMs have great potential to perform KBC tasks when prompted effectively. However, we still observe a list of limitations during the process: (1) The temporal information gap within the GPT-family of models may result in providing inaccurate responses. (2) The free-form of generation of generative PLMs makes the evaluation of the model's true capacity challenging. (3) Models struggle with the actual number of answers for relations such as `"PersonPlaysInstrument"` and potentially will hallucinate by returning answers that should not be returned. (4) We require humans to design the prompt template and hence will need to re-design when adapting to other tasks. In the future, different paths can follow to address current limitations.

1. Utilizing automatic prompt optimization techniques such as in [19]. Instead of human modifying prompts, we can learn the most optimal prompts automatically.
2. Chaining large language model prompts [20] to iteratively feed the output of the previous response to the next, aiming to amplify the advantages at each step and provide a more structured interaction with the model. Given this technique, we might be able to address the hallucination problem to some extent. A chain-of-thought prompt provides internal validation and improves the models' robustness in responses.
3. Exploring the effects of example selectors with different attributes. Currently, the example selector only considers the types of relations and the possible number of objects. Another avenue to explore is to select examples based on the properties of each relation type. Being able to understand how exactly example selectors affect the response could help to generalize to other tasks.

Acknowledgments

We thank the organizers of the 2023 Knowledge Prompting Hackathon[2] where this research was started. We also thank Data Language[3] for providing resources for our experiments. This research is supported in part by Dutch Research Council (NWO) through grant MVI.19.032, the European Union's Horizon 2020 research and innovation programme within the OntoTrans project (No. 862136) and the ENEXA project (grant agreement no. 101070305), as well as the Austrian Science Fund (FWF) within the HOnEst project (No. V 754-N).

Authors' contribution

Authors' contribution according to CRediT (https://credit.niso.org/): Investigation, Methodology, Conceptualization (XL, AH, ML, FP, FE); Software (XL, AH, ML, FP, FE); Supervision (PG, FE); Writing (XL, AH, ML, FP, PG, FE).

References

[1] S. Singhania, J.-C. Kalo, S. Razniewski, J. Z. Pan, Lm-kbc: Knowledge base construction from pre-trained language models, semantic web challenge @ iswc, CEUR-WS (2023). URL: https://lm-kbc.github.io/challenge2023/.

[2] H. Li, Language models: Past, present, and future, Commun. ACM 65 (2022) 56–63. URL: https://doi.org/10.1145/3490443. doi:10.1145/3490443.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[4] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, arXiv preprint arXiv:2204.02311 (2022).

[5] E. Beeching, C. Fourrier, N. Habib, S. Han, N. Lambert, N. Rajani, O. Sanseviero, L. Tunstall, T. Wolf, Open LLM Leaderboard, https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

[6] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[7] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation Extraction By End-to-end Language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204. doi:10.18653/v1/2021.findings-emnlp.204.

[8] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Celikyilmaz, Y. Choi, Comet: Commonsense transformers for automatic knowledge graph construction, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4762–4779.

[2] https://king-s-knowledge-graph-lab.github.io/knowledge-prompting-hackathon/

[3] www.datalanguage.com

[9] I. Melnyk, P. Dognin, P. Das, Grapher: Multi-stage knowledge graph construction using pretrained language models, in: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021.

[10] S. Singhania, T.-P. Nguyen, S. Razniewski, LM-KBC: Knowledge Base Construction from Pre-trained Language Models, in: S. Singhania, T.-P. Nguyen, S. Razniewski (Eds.), Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022, volume 3274 of *CEUR Workshop Proceedings*, CEUR, Virtual Event, Hanghzou, 2022, pp. 1–10. URL: https://ceur-ws.org/Vol-3274/#paper0.

[11] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. URL: https://aclanthology.org/D19-1250. doi:10.18653/v1/D19-1250.

[12] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, D. Roth, Recent advances in natural language processing via large pre-trained language models: A survey, ACM Comput. Surv. (2023). URL: https://doi.org/10.1145/3605943. doi:10.1145/3605943, just Accepted.

[13] D. Alivanistos, S. B. Santamaria, M. Cochez, J.-C. Kalo, E. v. Krieken, T. Thanapalasingam, Prompting as Probing: Using Language Models for Knowledge Base Construction, in: S. Singhania, T.-P. Nguyen, S. Razniewski (Eds.), Proceedings of the Semantic Web Challenge on Knowledge Base Construction from Pre-trained Language Models 2022, volume 3274 of *CEUR Workshop Proceedings*, CEUR, Virtual Event, Hanghzou, 2022, pp. 11–34. URL: https://ceur-ws.org/Vol-3274/#paper2.

[14] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, 2023. arXiv:2308.02357.

[15] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[16] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Computing Surveys 55 (2023) 1–35.

[17] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. arXiv:2203.02155.

[18] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences, Advances in neural information processing systems 30 (2017).

[19] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, CoRR abs/2010.15980 (2020). URL: https://arxiv.org/abs/2010.15980. arXiv:2010.15980.

[20] T. Wu, M. Terry, C. J. Cai, Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts, in: Proceedings of the 2022 CHI conference on human factors in computing systems, 2022, pp. 1–22.