

Enhancing Knowledge Base Construction from Pre-trained Language Models using Prompt Ensembles

Fabian Biester¹, Daniel Del Gaudio¹ and Mohamed Abdelaal²

¹University of Stuttgart, Stuttgart, Germany

²Software AG, Darmstadt, Germany

Abstract

Large language models such as ChatGPT and Bard manifest a significant step in the are of artificial intelligence. Yet, extracting useful knowledge from such models is still a challenging task. Due to the nature of language models, responses can be inaccurate, biased or even speculative. Predicting accurate object-entities by utilizing language model probing is the goal of the LM-KBC challenge. Our approach focuses on the concept of prompt ensembles. We employ initial baseline prompts to ChatGPT and then refine those prompts to exclude suboptimal ones. After a few shot learning step, we use prompt elicitation to improve the output. We use the Llama2 model with 70 billion parameters for inference. Our evaluation shows that this technique significantly enhances previous methods for knowledge base construction from language models. Our implementation is available on <https://github.com/asdfthefourth/lmkbc>.

1. Introduction

The advent of large models (LMs), e.g., OpenAI ChatGPT and Google Bard, marks a significant leap forward in the realm of artificial intelligence (AI) and human-computer interaction. Such LMs are able to comprehend and generate human-like text, enabling them to serve as unparalleled knowledge inventories. With their immense linguistic capacity and extensive training data, these models can swiftly process and provide information on a vast array of subjects.

While such models offer immense potential as knowledge inventories, extracting such knowledge remains a challenging task [1]. Specifically, extracting useful knowledge from these models requires careful consideration of their inherent limitations. The models' responses are generated based on patterns learned from vast datasets, which can lead to instances of inaccuracies, biases, and even the presentation of speculative information. The lack of a discerning mechanism to validate the accuracy of the information presents a significant hurdle in ensuring the reliability of the knowledge dispensed. Moreover, the models might struggle with context comprehension in complex or specialized domains, leading to responses that may seem plausible but lack depth.

In this paper, we seek to address this problem via introducing our innovative solution to the LM-KBC challenge [2]. The LM-KBC challenge aims to advance the field of knowledge base construction from pre-trained language models. It is required to develop solutions that


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

✉ st108056@stud.uni-stuttgart.de (F. Biester); Daniel.Del-Gaudio@ipvs.uni-stuttgart (D. Del Gaudio);

Mohamed.Abdelaal@softwareag.com (M. Abdelaal)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

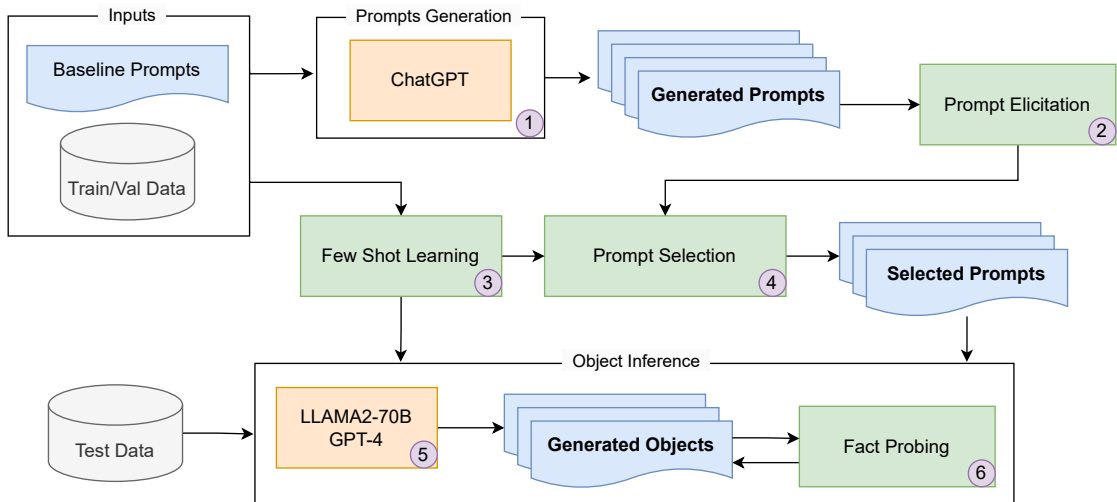


Figure 1: Overview of the proposed solution.

can construct knowledge bases using pre-trained language models. Specifically, the challenge entails the prediction of the accurate object-entities through the utilization of Language Model probing, with the provided input subject-entity and relation. The LM-KBC challenge dataset comprises a diverse set of 21 relations, each covering a different set of subject-entities, and a complete list of ground truth object-entities per subject-relation-pair.

Our solution primarily relies on the concept of prompt ensemble¹. Figure 1 depicts the architecture of our proposed solution, highlighting the various components. To craft the prompt ensemble, we initiate the process by employing the initial baseline prompts as inputs to the ChatGPT framework. Through this procedure, ten analogous prompts are generated, all of which pose the identical query. Subsequently, a step of prompt elicitation is executed, wherein supplementary contextual information is injected into the chosen prompts, furnishing them with additional insights about the anticipated nature of the predicted data.

Preceding the inference phase, a process of prompt refinement is undertaken to exclude suboptimal prompts, drawing from insights garnered from the train/validation dataset. The selected prompts of merit are then augmented with illustrative instances drawn from the training data, thereby facilitating the enhancement of few-shot learning capabilities. Furthermore, a validation process known as fact probing ensues, which assures the accuracy of the selected prompts. In this phase, a transformation of the relation is undertaken to assume the form of a Boolean question. For instance, for a relation like "BandHasMember," this transformation would manifest as the query, "Is member a constituent of band?".

The final stage involves the inference of object entities, using the Llama2 70B model, which engages its adeptness in disentangling complex data relationships. In the remainder of this paper, we will introduce these components in more detail, before discussing the obtained results.

¹Our implementation is available on <https://github.com/asdfthefourth/lmkbc>.

2. Prompt Ensemble

The main contribution of our work is using prompt ensembles for LM probing [3]. Instead of using a single prompt, a linguistic diverse set of prompts is used, i.e., an ensemble. We define an individual threshold per relation to reach consensus between the prompts. Additionally, the n worst prompts based on the performance on the training data are removed. We take the number x of possible answers from the dataset description and only use the number as the upper bound of possible answers for the model. We finetune the threshold on the validation dataset and use the resulting values for the test dataset. For choosing the optimal subset for the ensemble we calculated all subsets between 4 to all prompts on the training dataset and evaluated those on the validation dataset. The final prompt consists of the general context, the relation specific context, the few-shots and the prompt itself. The prompt ensemble approach is divided into five steps that are further explained in this section: prompt generation (Section 2.1), prompt elicitation (Section 2.2), few-shot example (Section 2.3), prompt selection (Section 2.4) and entity disambiguation (Section 2.5).

2.1. Prompt Generation

For generating additional prompts, we use the base prompt and ChatGPT to generate an additional x prompts. Also, additional prompts were checked by humans for consistency and errors. The idea behind using LMs for prompt generation is to exploit model understanding to get more similar questions.

2.2. Prompt Elicitation

In the prompt elicitation step, we add additional information to some of the relations [4]. This information is very specific for each relation.

For example, regarding the relation `PersonPlayInstrument`, we add the musician in front of the subject-entity. For the relation `BandHasMember`, we add the band in front of the subject-entity. One example for a prompt for the relation `BandHasMember` is "Who are the members of subject_entity?". After elicitation, the prompt looks like this: "Who are the members of the band subject_entity?". An example for the relation `PersonPlaysInstrument` is "What instrument does subject_entity play?", which looks like this after elicitation: "What instrument does the musician subject_entity play?".

Table 6, Table 7 and Table 8 in Appendix A show the worst and the best performing prompts resulting from our approach.

2.3. Few-Shot Example

We choose the few-shot examples randomly for each relation by using the training data and the base prompt provided in the dataset repo [5]. We test which amount of few-shots produces the best F1-score. Table 1 shows the results for a given few-shot number k .

Table 2 shows the similar results for only using the best prompt ensemble. The results show that the comparison of a higher amount of examples provides a better result. Yet, diminishing returns happen after $k = 10$ and the computation effort increases too much for using $k = 20$

Table 1

Comparisons of few-shot numbers for best prompt.

k	Precision	Recall	F1-Score
0	0.192	0.311	0.178
2	0.522	0.548	0.494
5	0.545	0.566	0.517
10	0.562	0.605	0.544
20	0.582	0.584	0.551

Table 2

Comparisons of few-shot number for the best prompt ensemble.

k	Precision	Recall	F1-Score
0	0.455	0.291	0.264
2	0.609	0.584	0.559
5	0.615	0.601	0.576
10	0.628	0.609	0.588
20	0.634	0.604	0.584

or higher. Using only the best prompt, the F1-score reduces even when using ensemble. the few-shot examples are handpicked to include the maximum size of answers, the minimum size of answers, the empty set if it is allowed and a diverse selection of answers if only a few answer classes are available.

2.4. Prompt Selection

Prompt selection is done by trying out a subset of ensembles on the training or the evaluation datasets and varying thresholds for finding a consensus. The best subset of prompts is chosen based on the F1-score.

We only search for thresholds in relations that allow more than one object and the empty set, because the consensus algorithm produces the same results regardless of the threshold. This is only the case for the two relations with numbers as solutions: `PersonHasNumberOfChildren` and `SeriesHasNumberOfEpisodes`.

2.5. Entity Disambiguation

We use the baseline entity disambiguation provided with the dataset and modify the algorithm to return an empty string if no entity was found. We prompt the LM in the few shot example to return the objects in the following format: `[Object1, Object2, ...]`. Then we extract the objects and query the Wikidata² API for IDs as it was done in the baseline.

²Wikidata: <http://wikidata.org>

3. Fact Probing

Fact probing is used to check whether the result of the previous step is correct [6]. For each relation, the LM is asked whether it is true or not. Therefore, we invert the relationship of each relation and create a prompt as a boolean question. For example, for every band in a relation {band} BandHasMember {member}, we create a prompt "is {member} part of {band}". In this step, we also do few shot examples to improve the LMs response. For example, regarding the relations related to death, we first ask whether the subject is still alive and then confirm either location or cause.

4. Experimental Setup

The development of our approach was mostly done on the Llama2 model with 13 billion parameters to improve the turnaround time.

The following steps are taken to achieve the results. We run the validation dataset on the Llama2 model with 70 billion parameters with 10 few-shot examples. Then, we do fact checking on the results. Next, we generate the ensembles and the thresholds for each relation based on the validation dataset. We run test dataset on Llama2 and GPT-4 while using the generated ensembles from the previous step. Lastly, we combine the final dataset.

The last step contains the combination of the final dataset, we combine the solutions that were generated by gpt4 and llama respectively and combine them to a single dataset.

4.1. Dataset Description

The dataset we use for the evaluation contains 21 m-to-n relations. Four of these relations contain the empty set as the solution. The dataset contains 1940 subjects for predictions and is divided into a training, validation and test dataset. The dataset contains relations in different domains, e.g., chemistry, geography and celebrities. The relations are in the triple-manner subject-predicate-object. An example for a relation in the chemistry dataset is: subject: "potassium, hydrogen, oxygen", relation: "CompoundHasParts", object: "Potassium Hydroxide". An example for an empty relation in the celebrity dataset is: subject: "Kobe Bryant", relation: "PersonHasNoblePrize", object: "" (empty set).

We use the training dataset for the few-shot training and the validation dataset for calculating the optimal ensemble.

4.2. Model Selection

We use the Llama2 model in several editions: with 7 billion parameters, with 13 billion parameters and with 70 billion parameters. We run the Llama2 model with 70 billion parameters on two A100 GPUs with 40GB RAM each for inference. We furthermore use a non-fine-tuned version and a fine-tuned version for the chat completion. Table 3 shows the results of our comparison of the Llama2 model with different parameter sizes, each using the best prompt and the best ensemble. Our comparison shows that the model version with 70 billion parameters using the best ensemble achieves the best results.

Table 3

Comparisons of the Llama2 model with different parameter sizes, each with the best prompt or the best ensemble.

Model version	Precision	Recall	F1-Score
7b with best prompt	0.537	0.535	0.505
7b with best ensemble	0.602	0.576	0.552
13b with best prompt	0.517	0.697	0.501
13b with best ensemble	0.621	0.592	0.562
70b with best prompt	0.663	0.628	0.585
70b with best ensemble	0.693	0.675	0.645

Table 4

Comparisons of the fine-tuned and the non-fine-tuned Llama2 model, each with the best prompt or the best ensemble.

Model version	Precision	Recall	F1-Score
fine-tuned with best prompt	0.556	0.538	0.523
fine-tuned with best ensemble	0.607	0.579	0.562
non-fine-tuned with best prompt	0.572	0.580	0.544
non-fine-tuned with best ensemble	0.634	0.609	0.589

Table 4 shows the results of the comparison of the fine-tuned and non-fine-tuned models, each using the best prompt and the best ensemble. The comparison shows that the non-fine-tuned model with the best ensemble achieves the best results.

We chose the 13 billion parameter non-fine-tuned model version for development due to our limited processing resources. For the final results, we used the 70 billion non-fine-tuned version. In both model versions, we applied a quantization on the weights from 16 bit to 4 bit to allow faster inference time and to allow the model to fit inside a single GPU instead of four. This leads to a loss in precision and, thus, a lower F1-score as it would be possible with our approach.

Furthermore, we use GPT-4 for the relation CountryHasStates, PersonHasSpouse, BandHasMember, because GPT4 is performing better on those relationships.

5. Results

Table 5 shows the results of our experiments using the validation dataset. We evaluated the precision, recall and F1-score and calculated the average over all relations. We achieved an average F1-score of **62.53%**.

It is noticeable that the prediction of some relations performs much better than the one of others. For example, our approach performs worst predicting PersonHasEmployer with an F1-score of 34.97% and performs best predicting the relation PersonHasNoblePrize with an F1-score of 98.00%. This might be due to the different kinds of datasets and the way LMs are trained. For example, winning a noble price or a chemical relation is a much more unique relation than having an employer.

Table 5

Final results of our approach on the validation dataset.

Relation	Precision	Recall	F1-Score
BandHasMember	0.6156	0.6414	0.5920
CityLocatedAtRiver	0.6900	0.6048	0.6099
CompanyHasParentOrganisation	0.8700	0.6150	0.5867
CompoundHasParts	0.9780	0.9755	0.9747
CountryBordersCountry	0.8248	0.8402	0.8038
CountryHasOfficialLanguage	0.8949	0.8346	0.8413
CountryHasStates	0.5770	0.7115	0.6214
FootballerPlaysPosition	0.6050	0.7433	0.6413
PersonCauseOfDeath	0.7000	0.7433	0.6950
PersonHasAutobiography	0.3417	0.4150	0.3547
PersonHasEmployer	0.4163	0.3777	0.3497
PersonHasNoblePrize	0.9900	0.9900	0.9800
PersonHasNumberOfChildren	0.5400	0.5400	0.5400
PersonHasPlaceOfDeath	0.5100	0.5500	0.5100
PersonHasProfession	0.3899	0.4978	0.3978
PersonHasSpouse	0.6800	0.6600	0.6633
PersonPlaysInstrument	0.6683	0.4353	0.4779
PersonSpeaksLanguage	0.9008	0.7702	0.7856
RiverBasinsCountry	0.8123	0.8529	0.7803
SeriesHasNumberOfEpisodes	0.4100	0.4100	0.4100
StateBordersState	0.5316	0.6012	0.5163
Average	0.6641	0.6576	0.6253

6. Related Work

Alivanistos et al. [6] focus on prompting as probing which is a multi-step approach that combines a variety of prompting techniques to construct knowledge bases from LMs. We apply a similar approach in the step of fact probing. Yet, we further improve their approach by using prompt ensembles in a previous step.

Li et al. [4] focus on the task-specific to improve relation prediction using LMs. Therefore, they create a sentence from every subject-relation-object triple. They mask the tokens in the sentence that are relevant to the object entity and train the LM with the objective to predict the tokens. They furthermore apply prompt elicitation, similar to our approach.

Ning and Celebi [3] also apply a prompt ensemble step, similar to our approach. Yet, we furthermore improve this by applying additional steps like prompt elicitation and fact probing.

In their pre-print “Boosted Prompt Ensembles for Large Language Models”, Pitis et al. [7] use few shot prompts to create prompt ensembles. They adapt classical boosting algorithms to improve prompts in an iterative process. Their solution could be a promising future improvement for our work.

Jiang et al. [8] propose a mining- and paraphrasing-based method to estimate the knowledge contained in LMs more accurately. Similar to our approach, they use ensemble methods to combine the results of different prompts. They describe different methods to create the ensembles,

which could be used to further improve our work. Yet, they do not involve ChatGPT for the prompts generation.

7. Conclusion and Outlook

Our evaluation shows that using prompt ensembles improves the overall performance of knowledge base construction using LMs. Our comparison described in Section 4.2 shows that using the non-fine-tuned Llama2 model with the most parameters achieves the best results. This leads to a trade-off between the best results and processing time.

In the future, we aim to evaluate our approach on more powerful hardware so we do not have to apply quantization to the model parameters. This should lead to an ever higher F1-score. We furthermore aim to test our approach of different LMs like GPT-4 and introduce the approaches of Ning and Celeb [7] and Jiang et al. [8] to our solution, as described in Section 6.

Acknowledgments

This research was funded by German Federal Ministry of Education and Research (BMBF) through grants 01IS17051 (Software Campus program), 02L19C155, 01IS21021A (ITEA project number 20219).

References

- [1] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, arXiv preprint arXiv:1909.01066 (2019).
- [2] S. Singhanian, 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/>.
- [3] X. Ning, R. Celebi, Knowledge base construction from pre-trained language models by prompt learning (2022).
- [4] T. Li, W. Huang, N. Papasrantopoulos, P. Vougiouklis, J. Z. Pan, Task-specific pre-training and prompt decomposition for knowledge graph population with language models, arXiv preprint arXiv:2208.12539 (2022).
- [5] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM Comput. Surv. 53 (2020).
- [6] D. Alivanistos, S. B. Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, T. Thanapalasingam, Prompting as probing: Using language models for knowledge base construction, arXiv preprint arXiv:2208.11057 (2022).
- [7] S. Pitis, M. R. Zhang, A. Wang, J. Ba, Boosted prompt ensembles for large language models, 2023. arXiv:2304.05970.
- [8] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How Can We Know What Language Models Know?, Transactions of the Association for Computational Linguistics 8 (2020) 423–438.

A. Worst and Best Performing Prompts

Table 6, Table 7 and Table 8 show the worst and best performing prompts for each relation.

Table 6

Worst and best performing prompts for each relation (first of three).

Relation	Prompt	F1-Score
BandHasMember	Who comprises the music group of subject_entity?	0.5325
BandHasMember	What are the band members' names in the case of subject_entity?	0.4801
CityLocatedAtRiver	Which river is subject_entity located at?	0.6129
CityLocatedAtRiver	Mention the river where subject_entity is situated.	0.473
CompanyHasParentOrganisation	What is the parent organization of subject_entity?	0.5266
CompanyHasParentOrganisation	Can you name the company that is the parent organization of subject_entity?	0.2316
CompoundHasParts	What are the components of subject_entity?	0.9784
CompoundHasParts	What is subject_entity made up of in terms of its components?	0.7464
CountryBordersCountry	Can you tell me the neighboring countries of subject_entity?	0.863
CountryBordersCountry	Tell me about the countries that are in close proximity to subject_entity.	0.8208
CountryHasOfficialLanguage	Can you tell me the language that serves as the official language of subject_entity?	0.9440
CountryHasOfficialLanguage	Which language is used for official purposes in subject_entity?	0.9081

Table 7

Worst and best performing prompts for each relation (second of three).

Relation	Prompt	F1-Score
CountryHasStates	Provide a list of states that belong to subject_entity.	0.0
CountryHasStates	What are the constituent states of subject_entity?	0.0
FootballerPlaysPosition	Can you tell me in which position subject_entity participates in football?	0.655
FootballerPlaysPosition	Tell me about the role subject_entity plays in football.	0.5566
PersonCauseOfDeath	What caused the death of subject_entity?	0.7783
PersonCauseOfDeath	Share details about what led to subject_entity's passing.	0.5366
PersonHasAutobiography	Mention the title of the book authored by subject_entity.	0.4266
PersonHasAutobiography	I'm interested in knowing the title of subject_entity's autobiography.	0.3166
PersonHasEmployer	Who is subject_entity's employer?	0.3027
PersonHasEmployer	Can you tell me the name of subject_entity's employer?	0.2142
PersonHasNoblePrize	In which field did subject_entity receive the Nobel Prize?	0.9766
PersonHasNoblePrize	What discipline was recognized when subject_entity was awarded the Nobel Prize?	0.5966
PersonHasNumberOfChildren	How many children does subject_entity have?	0.52
PersonHasNumberOfChildren	What is the size of subject_entity's family in terms of children?	0.39

Table 8

Worst and best performing prompts for each relation (third of three).

Relation	Prompt	F1-Score
PersonHasPlaceOfDeath	Where did subject_entity die?	0.4444
PersonHasPlaceOfDeath	Identify the location of subject_entity's passing.	0.303
PersonHasProfession	What does subject_entity do as their profession?	0.368
PersonHasProfession	State subject_entity's area of expertise or job title.	0.3079
PersonHasSpouse	Who is the life partner or spouse of subject_entity?	0.6383
PersonHasSpouse	Provide information about the person to whom subject_entity is married.	0.13
PersonPlaysInstrument	What is subject_entity's chosen musical instrument?	0.5555
PersonPlaysInstrument	Tell me about the musical equipment with which subject_entity is proficient.	0.43
PersonSpeaksLanguage	What languages does subject_entity speak?	0.8279
PersonSpeaksLanguage	What are the different languages known by subject_entity?	0.7376
RiverBasinsCountry	State the country where the subject_entity river basin is situated.	0.8382
RiverBasinsCountry	I'd like to know the geographical location of the subject_entity river basin.	0.7951
SeriesHasNumberOfEpisodes	How many episodes have been produced for subject_entity series?	0.5
SeriesHasNumberOfEpisodes	Tell me about the total episode tally of subject_entity series.	0.43
StateBordersState	Tell me the names of the states that share borders with subject_entity.	0.4643
StateBordersState	What states are connected to subject_entity's state by borders?	0.3912