Limits of Zero-shot Probing on Object Prediction

Shrestha Ghosh[1,2]

[1]Max Planck Institute for Informatics, Saarbrücken 66125, Germany
[2]Saarland University, Saarbrücken 66125, Germany

Abstract

In this work, we present Minimal Probe. The system is one of the participants of the LM-KBC challenge 2023, which tackles the task of Knowledge Base Construction using Large Language Models (LLMs). Since LLMs are trained on huge amount of general knowledge, they are known to store knowledge in their parameters. They have been probed for factual and commonsense knowledge. Minimal Probe aims to analyze how LLMs perform in low-resource setting. By careful prompt construction and intuitive answer cleaning, we show that LLMs can be used to extract multiple objects for a given subject and relation, without any demonstrations. Our system performs equally well on precision and recall, surpassing the baseline by more than 40% on F1. Minimal Probe achieves an average F1 score of 0.608 on the hidden test set: only 9.2% behind the winning team, which does use demonstrations.
The code and results are available at https://github.com/ghoshs/LM-KBC2023.

1. Introduction

Extracting knowledge from Large Language Models (LLMs) for Knowledge Bases (KBs) has been tackled in recent literature [1]. Predicting facts for entities where the relation cardinality is not fixed is yet under-explored. LLMs for natural language processing tasks are more popular with demonstrations-aided prompting [2, 3] and have been shown to exhibit emergent abilities, such as performing tasks on few-shot prompts, with increasing parameters [4]. Newer models trained on code and text are more capable of generating structured data [5, 6].

Prompts consist of the task description and optional demonstrations instantiating the task input/output, followed by the input for which the model is expected to provide an answer. The prompting technique with no demonstrations is called *zero-shot* prompting, and when demonstrations are provided, it is called a *few-shot* or *k-shot* prompting. The zero-shot setting is interesting since it allows us to determine how much knowledge is already stored in LLMs and how effectively could we retrieve this parameterized knowledge. Previous works have shown LLMs to be good zero-shot reasoners [7].

1.1. LM-KBC Challenge

This is the second edition of the LM-KBC challenge, which addresses the task of KB construction using LLMs [8]. Here, the LLMs provide an alternative to the traditional information extraction from unstructured text. As in the first edition [9], the main task is to predict all objects given

KBC-LM'23: Knowledge Base Construction from Pre-trained Language Models workshop at ISWC 2023
ghoshs@mpi-inf.mpg.de (S. Ghosh)

a subject and a relation. Since the number of objects is not fixed and could be zero as well, using LLMs is especially challenging. New additions to the current challenge included entity disambiguation to Wikidata and predicting relation cardinality as an integer.

Two major observations from the first edition of the LM-KBC challenge [9] are that larger models have higher performance and that triple-based prompts performed better than natural language [10].

1.2. Related Work

The potential of LLMs as KBs was first explored by Petroni et al. [1], where they introduced the LAMA probe to test knowledge stored in LLMs. This prompted a recent body of work on probing LLMs for factual knowledge and KB curation tasks [11, 12, 13, 14, 15, 16]. There are still challenges to using LLMs for and as KBs which need more focus [17].

Prompting is an important part of using LLMs for downstream tasks [18]. Zero-shot prompting has received some attention in recent literature for task generalization [19] and chain-of-thought prompting [7]. Early experiments with GPT-4 have shown remarkable increment on certain tasks with zero-shot prompts as compared to the previous models [20].

2. Approach

Through this challenge, this work investigates how much information can be retrieved from LLMs using zero-shot prompts on the GPT family of models. Through our system, named Minimal Probe, we will focus on the following aspects: *prompt design*, and *answer post-processing*.

2.1. Prompt Design

The prompt consists of three parts: i) *a task description*, ii) optional *demonstrations*, and, iii) *the task* itself. We focus on each section one by one.

Task description. The challenge presents a baseline (GPT3 (curie-002) + NED) with no task description and only demonstrations, which achieves 21% F1. We hypothesize that a task description provides a guide for the model to output more reliably and formulate a simple task description where the models asked exactly what is required in the output, *i.e.,* a list of values.

Example 1. *A prompt with the task description on the first line and the task in the next line.*

> *Please fill the empty list, if necessary, to create a correct fact. Return a valid tuple.*
> *("Paraguay", "borders country", [])*

We focus on the task description as an alternative to providing demonstrations and probing LLMs in a zero-shot way.

Demonstrations. Demonstrations serve the purpose of laying the format of the task. As mentioned earlier, with just some sample demonstrations in the prompt, GPT3 (curie-002) achieves an F1 score of 21%. In the case where an LLM is prompted directly with the task without any task description or demonstrations, the output structure becomes very unreliable. The output could be anything from a comma-separated string to a paragraph about the subject entity. This is probably because the model has not seen enough text, if not any, in the prompt format to make predictions in a consistent format. This work probes the limits of LLMs for KB construction without using demonstrations.

Task. The task itself can be divided into two components: the *format* and the *content*. As shown in Example 1, the model is provided with an instruction to *fill an empty list* in a tuple format. Editing and inserting capabilities were introduced to the GPT family of LLMs in March 2022 [21] and is carried forward in to the newer models. The format guides the LLM to return a similar response. As a result, the predictions can be automatically parsed with minimal effort.

Another departure from the baseline is in presenting the relation. Even though LLMs returns acceptable answers when probed with the provided relation as is, *i.e.,* in camel-case, we probe the model with relations in natural language, but without the subject type to see how well the model performs. The relations in their current form have certain inconsistencies. Some object types are in the singular. `CountryBordersCountry` and some in the plural: `Country-HasStates`. The relation `RiverBasinsCountry` has no verb. We paraphrased the relations to have plural object type and added missing verbs. Some relations, which were more generic, returned results of granularity different from those expected in the evaluation. For instance, the relation `CompoundHasParts` was changed to more specific, `has elements`. The relations `CountryHasStates` and `StateBordersState` were modified such that the object type "*state*" was replaced by "*provinces*".

2.2. Answer Post-processing

Extraction. The careful prompt design keeps LLM response from deviating from the expected output. As a result, minimal parsing is required to extract the object entities. The response is parsed by the abstract syntax tree library of Python[1] using its `literal_eval` function. In case of a parse error, a regex pattern is used to extract all word groups within double quotes. We keep all matched patterns which do not match the task subject and relation. Once we have all the surface forms, except for the integers, we use the entity search service of the MediaWiki action API[2] to link the surface forms to Wikidata entities.

Cleaning. Another pass of processing is required after linking surface forms to Wikidata entities to make the object entities compatible with the desired format of the challenge.

Firstly, integer objects are encoded as strings. *Secondly*, if an object is not an integer, we check if non-entity literals, such as "*unknown*", "*N/A*", "*none*", "*false*" and its variants have been predicted. If yes, then the corresponding object IDs if any are removed since the current

[1] https://docs.python.org/3/library/ast.html

[2] We use the `wbsearchentities` action. API page: https://www.mediawiki.org/wiki/API:Main_page

evaluation treats them as empty values. *Thirdly*, object IDs which are the same as the subject IDs are removed. Since none of the relations are reflexive, this operation can be safely performed.

Next, a relation specific clean-up is applied to specific relations. For the `PersonHasNobelPrize` relation, it was observed that the majority of the answers returned were the fields of the award, which was not enough to disambiguate the objects. Since the object range of Nobel Prizes comprises six awards, we mapped each category's surface form (the field names) to its Wikidata ID. Whenever the named-entity disambiguation failed, *i.e.,* the object entity did not match any of the six award entities, we directly looked up the Wikidata ID from its object surface. This is a pragmatic choice when the object range is limited. In the case of `CityLocatedAtRiver`, whenever the object returned did not belong to the class river, we re-ran the KB linking search by appending the surface form with the object type, here *river*.

3. Results

The main results on the validation set are presented in Table 1 and Table 2. Based on these results, we determine the final configuration to be used on the test set.

Setup. We compare the performance of two models trained on human feedback: `GPT-3.5-turbo` and `GPT-4` [22]. The model parameters used are temperature set to 0, and max tokens set to 200.

Larger Models. As observed in the previous edition of the challenge, increasing the model size increased the overall performance. To confirm this, we evaluate the LM-KBC baseline on bigger models, without making any other changes. Table 1 shows the performance of the LM-KBC baseline increases when run with larger models. There is a big jump within the completion models when switching from `text-curie-001` to `text-davinci-003`, with a gain of more than 20% in F1. This stabilizes to about 5%-10% for higher models.

Answer cleaning. The cleaning component of answer post-processing improves the F1 by more than 10% in our Minimal Probe. This shows that the model outputs can still be noisy and just employing a larger model is not enough. Relation-specific cleaning of the objects, as was done for the Nobel Prize categories, increased the precision to 99%. Another observation was that the objects for parent organization of companies returned the companies itself, especially when the answer was None.

The cleaning component, which deals with normalizing unknown values to an empty value, also boost the performance. It is interesting to note that the LLMs distinguish between not knowing, *i.e., "unknown"* and knowing that there are no objects, *i.e., "none"* or *"false"*. However, for this challenge, both the meanings are merged to empty value.

Relation Modifications. In Table 2, we highlight how GPT-4 performs when the relations are presented i) in their original camel-cased format as presented in the challenge, ii) in natural language format without the subject type and, iii) after paraphrasing. We observe F1 increases for most relations after modifications. For those relations where the original scores the highest,

Table 1
Precision (P), Recall (R) and F1 of different systems on validation dataset.

System	Setting	P	R	F1
lmkbc2023-gpt-ned-baseline	text-curie-001 *(baseline)*	0.307	0.210	0.217
	text-davinci-003	0.493	0.533	0.437
	+ answer cleaning	0.512	0.549	0.476
	gpt-3.5-turbo	0.564	0.535	0.483
	+ answer cleaning	0.543	0.539	0.505
Minimal Probe **(ours)**	gpt-3.5-turbo			
	+ task description	0.493	0.478	0.419
	+ modified relations	0.512	0.529	0.441
	+ answer cleaning	0.539	0.569	0.523
	gpt-4			
	+ task description	0.541	0.554	0.502
	+ modified relations	0.525	0.591	0.508
	+ answer cleaning	**0.650**	**0.654**	**0.630**

`PersonHasAutobiography` and `CountryHasOfficialLanguage`, the other configurations are not too far behind. The most impacted is the relation `CompoundHasParts`, where the F1 increases by 50% points. For the test set, we keep the relations in their natural language format, but perform paraphrasing only for certain relations.

4. Discussion

In this section, we present some error analysis on the validation data and provide anecdotes on the pros and cons of using LLMs for KB curation. We observe that the majority of errors stem from relation-specific idiosyncrasies. In general, the relations dealing with people's professions and employers were most difficult to predict.

4.0.1. Granularity

With no other information except the task description and the task itself, it is sometimes difficult to control the granularity of the type of the objects returned. Wikidata provides property constraints such as value-type to restrict the class of objects. For instance, the objects for the country of citizenship relation is always of the type *country*, but it is not always the case. This was especially true for the relation `PersonHasProfession`, where the model predicts {"*actor*", "*producer*", "*director*"}, which can be considered sibling occupations, but the ground truth has "*actor*" as well as other subclass occupations, like *"film actor"*, *"television actor"*, *"voice actor"* and so on. These are particularly difficult to elicit unless provided with additional context.

Another example is the relation `CompanyHasParentOrganisation`, where the model often returned objects linked to the subject via the Wikidata relation *"owned by"*, which semantically is correct but not considered in the ground truth labels. Interestingly, the model never predicts "voice" as an object for the relation, `PersonPlaysInstrument`. `CityLocatedAtRiver` also suffers from the problem of granularity, sometimes returning water bodies instead of or along

Table 2
F1 scores by relations on GPT-4 for different relation configurations.

Relation	Original	Spaced-out	Paraphrased
BandHasMember	0.565	0.569	**0.583**
CityLocatedAtRiver	0.668	0.681	**0.695**
CompanyHasParentOrganisation	0.582	**0.622**	0.562
CompoundHasParts	0.443	0.449	**0.951**
CountryBordersCountry	0.774	0.773	**0.783**
CountryHasOfficialLanguage	**0.945**	**0.945**	0.941
CountryHasStates	0.275	0.326	**0.327**
FootballerPlaysPosition	0.628	**0.652**	0.648
PersonCauseOfDeath	0.700	0.730	**0.737**
PersonHasAutobiography	**0.467**	0.453	0.463
PersonHasEmployer	0.307	**0.341**	0.313
PersonHasNobelPrize	0.937	**0.950**	0.947
PersonHasNumberOfChildren	0.530	**0.540**	0.500
PersonHasPlaceOfDeath	0.515	**0.535**	0.505
PersonHasProfession	0.282	**0.322**	0.312
PersonHasSpouse	0.751	0.721	**0.771**
PersonPlaysInstrument	0.565	**0.575**	0.557
PersonSpeaksLanguage	0.810	**0.824**	0.803
RiverBasinsCountry	0.829	**0.837**	0.809
SeriesHasNumberOfEpisodes	0.510	**0.530**	0.520
StateBordersState	0.489	0.485	**0.501**
Average	0.599	0.612	**0.630**

with rivers.

4.0.2. Cardinality

The LLMs get almost 50% of the cardinalities wrong for both the relations `PersonHasNumberOfChildren` and `SeriesHasNumberOfEpisodes`. In the former relation, 50% of the error is one-off, for the latter though no visible pattern to the errors.

4.0.3. Recency

The relation `PersonHasEmployer` can be misleading for the LLM since it returns only recent employers, whereas the expectation is all employers, including past ones.

4.0.4. Ground Truth Mismatch

Very rarely does the model return an answer which doesn't match the ground truth, but can be corroborated in Wikidata or Wikipedia. One such triple is regarding the number of episodes for the Perry Mason series, which the model predicts as 271 as opposed to the ground truth prediction of 15. Another example is of the company AGL Resources, which has outdated

information on Wikidata and more recent information, including its parent organization in Wikipedia[3]. These instances are rare and generally due to changes not yet reflected in Wikidata.

4.0.5. Object Type

The relations most affected by disambiguation errors were `PersonHasNobelPrize` and `CityLocatedAtRiver`, where the Wikidata object returned were of types very different from the type expected for the relation. For instance, the Wikidata objects for the Nobel Prize were not of the type of award, but of the branch of science. In the instance of a city located at rivers, a common error occurred when the river name coincided with the name of a city and the Wikidata object returned was of the type city. There was a drastic increase in the F1 score, by more than 20% points, once this was handled.

5. Conclusion

The LM-KBC challenge explores the problem of KB construction using LLMs. Through our system, we show that LLMs not only improve with size, but are also capable of predicting under low-resource setting, such as zero-demonstrations. We show additionally that answer post-processing and relation-specific modifications can greatly improve fact prediction. Minimal Probe improves the baseline F1 by 42%, while maintaining the average precision and recall across relations at 65%. According to the official leaderboard on the hidden test set, Minimal Probe was only 9.2% points behind the winning system, which relied on demonstrations for final predictions.

Future work would include exploring whether further meta information such as object granularity or relation cardinality, such as the average number of objects, can improve object recall. Further, as evident from Table 2, relation surface forms are important. Worse representations may lead to poor predictions. Additionally, verification of the predicted objects remains a challenge. There are different design choices to be made as well: on one hand LLM as a retriever is less expensive, but answer verification is computation-heavy, on the other hand, LLM as an information extractor is expensive and limited by the model's context.

Acknowledgments

I thank the reviewers for their helpful suggestions.

References

[1] 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), 2019, pp. 2463–2473.

[3] https://en.wikipedia.org/wiki/Southern_Company_Gas

[2] T. Schick, H. Schütze, Few-shot text generation with natural language instructions, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 390–402.
[3] T. Schick, H. Schütze, It's not just size that matters: Small language models are also few-shot learners, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 2339–2352.
[4] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent abilities of large language models, Transactions on Machine Learning Research (2022). URL: https://openreview.net/forum?id=yzkSU5zdwD, survey Certification.
[5] A. Madaan, S. Zhou, U. Alon, Y. Yang, G. Neubig, Language models of code are few-shot commonsense learners, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022, pp. 1384–1403.
[6] X. Wang, S. Li, H. Ji, Code4struct: Code generation for few-shot event structure prediction, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 3640–3663.
[7] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, Y. Iwasawa, Large language models are zero-shot reasoners, Advances in neural information processing systems 35 (2022) 22199–22213.
[8] 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/.
[9] S. Singhania, T.-P. Nguyen, S. Razniewski, Lm-kbc: Knowledge base construction from pre-trained language models, CEUR-WS (2022). URL: https://lm-kbc.github.io/challenge2022/.
[10] 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, in: 2022 Semantic Web Challenge on Knowledge Base Construction from Pre-Trained Language Models, LM-KBC 2022, CEUR-WS.org, 2022, pp. 11–34.
[11] T. Safavi, D. Koutra, Relational world knowledge representation in contextual language models: A review, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 1053–1067.
[12] R. Zhao, F. Zhao, G. Xu, S. Zhang, H. Jin, Can language models serve as temporal knowledge bases?, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 2024–2037.
[13] S. Hao, B. Tan, K. Tang, B. Ni, X. Shao, H. Zhang, E. Xing, Z. Hu, Bertnet: Harvesting knowledge graphs with arbitrary relations from pretrained language models, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 5000–5015.
[14] 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.
[15] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4222–4235.
[16] B. Veseli, S. Singhania, S. Razniewski, G. Weikum, Evaluating language models for knowl-

edge base completion, in: European Semantic Web Conference, Springer, 2023, pp. 227–243.
[17] S. Razniewski, A. Yates, N. Kassner, G. Weikum, Language models as or for knowledge bases, arXiv preprint arXiv:2110.04888 (2021).
[18] 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.
[19] C. Zhou, J. He, X. Ma, T. Berg-Kirkpatrick, G. Neubig, Prompt consistency for zero-shot task generalization, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 2613–2626.
[20] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al., Sparks of artificial general intelligence: Early experiments with gpt-4, arXiv preprint arXiv:2303.12712 (2023).
[21] M. Bavarian, A. Jiang, H. Jun, H. Pondé, New gpt-3 capabilities: Edit & insert, 2022. URL: https://openai.com/blog/gpt-3-edit-insert.
[22] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.