Knowledge Base Construction from Pre-trained Language Models by Prompt learning

Xiao Ning[1,2], Remzi Celebi[2]

[1]School of Biological Science and Medical Engineering, Southeast University, China
[2]Institute of Data Science, Faulty of Science and Engineering, Maastricht University, the Netherlands

Abstract

Pre-trained language models (LMs) have advanced the state-of-the-art for many semantic tasks and have also been proven effective for extracting knowledge from the models itself. Although several works have explored the capability of the LMs for constructing knowledge bases, including prompt learning, this potential has not yet been fully explored. In this work, we propose a method of extracting factual knowledge from LMs for given subject-relation pairs and explore the most effective strategy to generate blank object entities for each relation of triples. We design prompt templates for each relation using personal knowledge and the descriptive information available on the web such as WikiData. The probing approach of our proposed LMs is tested on the dataset provided by the International Semantic Web Conference (ISWC 2022) LM-KBC Challenge. To cope with the problem of varying performance for each relation, we designed a parameter selection strategy for each relation. Using the test dataset, we obtain an F1-score of 0.4935%, which is higher than the baseline of 31.08%.

Keywords

Prompt learning, Pre-trained language model, Information Extraction, Link Prediction

1. Introduction

In recent years, the primary role of pre-trained language models (LMs) has transitioned from that of generating or evaluating the fluency of natural language text to being a powerful tool for understanding natural language [1, 2]. Pre-trained language models can be used as a knowledge base by formulating queries in natural language and either generating textual answers directly or assessing multiple choices and picking the most likely one. Regardless of the end task, the knowledge contained in LMs is probed by providing a prompt and letting LMs either generate the continuation of a prefix or predict missing words in a cloze-style template. A more direct approach to eliciting knowledge from these models is *prompting*, in which natural language prompts are used to query LMs, and the word assigned the highest probability in the blank will be returned as the answer [3, 4]. This task is highly related to link prediction in knowledge graphs, which plays an important role in knowledge graph construction.

Numerous techniques have been proposed to elicit such knowledge by analyzing LMs internal representations. Zhengbao Jiang et al. aimed to extract the knowledge contained in LMs by automatically discovering better prompts to use in the querying process [5]. Taylor Shin et al. developed an *automated* method named AUTOPTOMPT to create prompts for a diverse set of

LM-KBC'22: Knowledge Base Construction from Pre-trained Language Models, Challenge at ISWC 2022
x.ning@maastrichtuniversity.nl (X. Ning); remzi.celebi@maastrichtuniversity.nl (R. Celebi)

tasks based on a gradient-guided search [6]. Prompt learning does not require a large amount of labeled data or introduce a large number of additional parameters, which leads to a more useful analysis tool and has been widely used in many domains, such as name entity recognition [7], information extraction [8], question answer [9]. Nevertheless, prompting requires the manual design of the context to feed into the model, designing efficient prompt templates directly affects the performance of the model.

In this work, we develop a system for track 1 of the LM-KBC challenge, a challenge that aims to explore the viability of knowledge base construction from BERT [1] with low computational requirements. We propose an automatic method to systematically improve the performance of the prompts used to query the relations from pre-trained model. Our method is based on *bert-large-cased* [1] due to existing studies demonstrating its outstanding performance. This method is based on mining or paraphrasing that takes one prompt feed to the model. Considering that different prompts may have performance differences when used to query different relations, we also combined answers from different prompts together. The data, code and learned models associated with this work can be accessed in the Github repository [2].

2. Prompt Generation

We define prompt generation as the task of generating a set of prompts $t_{r,i}{}_{i=1}^{T}$ for each relation *r*, where at least some of the prompts effectively trigger LMs to predict ground-truth object-entities. Our method is inspired by template-based relation extraction methods, which are based on the observation that words in the vicinity of the subject s and object *o* in a large corpus often describe the relation *r*. We got the corresponding alternative description for each relation from the descriptive information in WikiData. Inspired by a template-based approach for relation extraction, we created prompt templates based on different descriptive information combined with professional knowledge. The three main method [5] we used in this challenge are below.

Middle-word Prompt Based on the observation that words in the middle of the subject and object are often indicative of the relation, we directly use those words as prompts. For example, *Sergey Brin set up Google* is converted into a prompt *s set up o* by replacing the subject and object with placeholders. For *CountryBordersWithCountry* relation, we design *" $\{subject\_entity\}$ shares border with $\{mask\_token\}$."* as one prompt.

Dependency-based Prompt In cases of templates where words do not appear in the middle, templates based on syntactic analysis of the sentence can be more effective for relation extraction tasks [10]. For instance, the dependency path in *The capital of China is Beijing* giving a prompt of *capital of s is o*. For *CompanyParentOrganization* relation, we designed *"The parent organisation of $\{subject\_entity\}$ is $\{mask\_token\}$."* as one prompt.

Paraphrasing-based Generation To improve lexical diversity while remaining relatively faithful to the original prompt, we paraphrased the original prompt with other semantically similar or identical expressions. When the prompt is *s shares a border with o*, it may be paraphrased as *s borders with o* and *s is next to o*. This is conceptually similar to the query expansion

[1] https://huggingface.co/bert-large-cased

[2] https://github.com/xiao-nx/LMKBC_2022

techniques which are used in information retrieval to reformulate a given query to improve retrieval performance [11].

3. Prompt Selection and Ensemble

In the previous section, we describe methods of generating a set of candidate prompts $\{t_{r,i}\}_{i=1}^{T}$ for a particular relation *r*. Each of these prompts may be more or less effective in eliciting knowledge from the LMs, and thus it is necessary to decide how to use these generated prompts during the test. In this section, we discuss the approaches explored for generating better candidate objects by prompt-based link-prediction. Our efforts here can be broadly classified into two categories: using better prompts and ensemble the prompts.

3.1. Selection of the Top-k Prompts

To find the prompts which better elicit the pre-trained model better, we designed prompts considering both a priori knowledge and synonyms as potential prompts. For each prompt, we can measure its precision, recall and F1-score of predicting the ground-truth objects on the training data, and keep several the top-performing prompts based on F1-score.

3.2. Ensemble Prompts

We do not observe the same scale of improvement with increasing number of prompts involved; in fact, most of the time the best F-1 score is achieved with one prompt template. We argue that this difference is due to the difference in the evaluation metrics: we pay attention to the F-1 scores rather than the macro-averaged accuracy scores, which give higher importance to the precision of methods. Therefore, considering that having a variety of prompts may allow for elicitation of knowledge that appeared in these different contexts, we rank all the prompts based on their performance of predicting the objects in the training set and keep the prompts with an F1-score higher than 0.1 or top 5. Although treating the top-k prompts equally is sub-optimal as some prompts are more reliable than others.

For every *relation* in the dataset, we use all filtered prompts to query the pre-trained language model, and every prompt will return a set of object entities. Then it is important to select the most accurate object entities. Here, we developed an algorithm that considers synthetically the frequency and probability of each predicted object-entities, and finally keep the top 5 candidates. Note that there often exist pronouns, such as *him, them, it*, or determiners, such as *the, a, any* in the top predicted objects, or other symbols, such as *?, 1970s, -s*, so we removed these words. In addition to that, we mapped the *music* in the predicted result into *producer*, *acting* into *actor*, *teacher* into *professor*, *water* into *hydrogen.*

4. Experiments

4.1. Dataset

The dataset for this challenge is divided into a training data, development data and test data, each covering a different set of subject-entities and along with complete list ground-truth object-entities per subject-relation-pair. The training data subject-relation-object triples can be used for training or probing the language models in any form, while development can be used for hyper-parameter tuning, and the test data is used to measure the performance of the final submitted system. Our proposed method is free from finetuning, so we just use the training data to test the performance of system tool and adjust parameters manually, then submitted the developed system tool.

4.2. Experimental Settings

Single Prompt Experiments For each prompt we designed, its corresponding performance was tested on the training set. The performance of top-3 is shown in Table 1.

Ensemble Prompts Experiments For some relations with low recall, we combined several prompts and rank them as the final results to get more object entities. We labeled the top 3 prompts as *prompt1, prompt2, prompt3*, and tried to evaluate the performance of ensemble *[prompt1, prompt2, prompt3], [prompt1, prompt1], [prompt1, prompt3], [prompt2, prompt3]* prompts and then took the best performing combination on the training data.

Search Threshold Experiments Another observation is that the threshold strongly affects the recall of the prediction results, and it is possible to obtain more object entities by lowering the threshold. Thus, we searched various thresholds to optimize the F-1 scores, and select the best thresholds based on the training data. According to the formula of F1 score, it is known that the F1 score achieves its maximum value when the accuracy and recall are close to each other, therefore we adjusted the threshold to search for the F1 score of optimal performance. Actually, in our experiments we performed only a small range of searching, but in order to show the effect of threshold on F1 socore clearly, we search the thresholds between 0.01 and 0.99 by steps of 0.01 and plotted in the Figure 1.

System Tool In this section, We present the prompt or the combination of prompts used for each relation and the corresponding threshold value, as shown in Table 2.

4.3. Final Test Results

As for the models to probe, in our main experiments, we use the BERT-large models. We use three metrics to evaluate the success of the prompts in probing LMs, precision, recall, and F1-score. The final performance of our proposed method on the test data can be seen in Table 3, as recorded on CodaLab [3].

[3] https://codalab.lisn.upsaclay.fr/competitions/5815

Table 1
The performance of each prompt on training data.

Prompts	Performance(t=0.1)			Performance(t=0.5)		
	Precison	Recall	F1-score	Precison	Recall	F1-score
s consists of m, which is an element.	0.302	0.565	0.289	0.12	0.97	0.09
s composed of m.	0.087	0.407	0.077	0.025	0.98	0.02
s comprised of m.	0.101	0.402	0.096	0.022	0.99	0.018
The parent organization of s is m.	0.411	0.625	0.66	0.6	0.94	0.63
s owed by m.	0.55	0.9	0.61	0.6	1.00	0.6
s is part of m.	0.187	0.312	0.64	0.54	0.89	0.61
s shares border with m.	0.458	0.737	0.453	0.13	0.98	0.118
s borders m.	0.403	0.757	0.399	0.109	1.00	0.099
s bordered by m.	0.451	0.723	0.446	0.146	0.98	0.132
The official language of s is m.	0.715	0.796	0.718	0.666	0.9	0.617
The language spoken in s is m.	0.658	0.72	0.69	0.488	0.91	0.463
The language official in s is m.	0.712	0.751	0.763	0.653	0.94	0.609
s died due to m.	0.042	0.057	0.515	0.43	0.94	0.48
s died of m.	0.037	0.029	0.54	0.11	0.46	0.48
s died from m.	0.04	0.032	0.54	0.14	0.55	0.48
s is the CEO of m.	0.02	0.883	0.023	0.025	0.98	0.023
s was appointed as CEO of m.	0.023	0.803	0.023	0.02	0.97	0.02
s is the chairman of m.	0.025	0.55	0.023	0.025	0.98	0.023
s plays m, which is an instrument.	0.414	0.43	0.729	0.259	0.99	0.266
s plays m.	0.274	0.56	0.38	0.244	1.00	0.242
s plays instrument of m.	0.112	0.24	0.255	0.18	0.55	0.24
s speaks in m.	0.67	0.633	0.865	0.43	0.98	0.406
s writes in m.	0.662	0.792	0.734	0.152	0.97	0.139
s second language is m.	0.621	0.567	0.831	0.143	0.99	0.13
s died at m.	0.222	0.478	0.53	0.48	0.98	0.49
s died in m.	0.32	0.435	0.55	0.46	0.98	0.48
The death place of s at m.	0.313	0.527	0.55	0.47	0.97	0.48
s is a m by profession.	0.042	0.155	0.029	0.005	0.99	0.003
s worked as a m.	0.034	0.533	0.02	0.002	1.00	0.001
s received a specialized professional training and became a m.	0.098	0.463	0.069	0.003	1.00	0.002
s river basins in m.	0.462	0.574	0.487	0.286	0.93	0.262
the watershed in m is s.	0.349	0.603	0.345	0.161	0.90	0.145
s borders m.	0.126	0.242	0.099	0.00	0.87	0.00
s bordered by m.	0.138	0.26	0.123	0.004	0.91	0.002
s adjacent to m.	0.117	0.266	0.105	0.00	0.91	0.00

s denotes $\{subject\_entity\}$, o denotes $\{mask\_token\}$.

5. Conclusion

Prompt learning exploits the powerful capability of pre-trained language models, and significantly minimizes the dependence on supervised data. Prompt learning enables shot learning and even zero shot learning, which is a promising application for NLP downstream tasks, especially

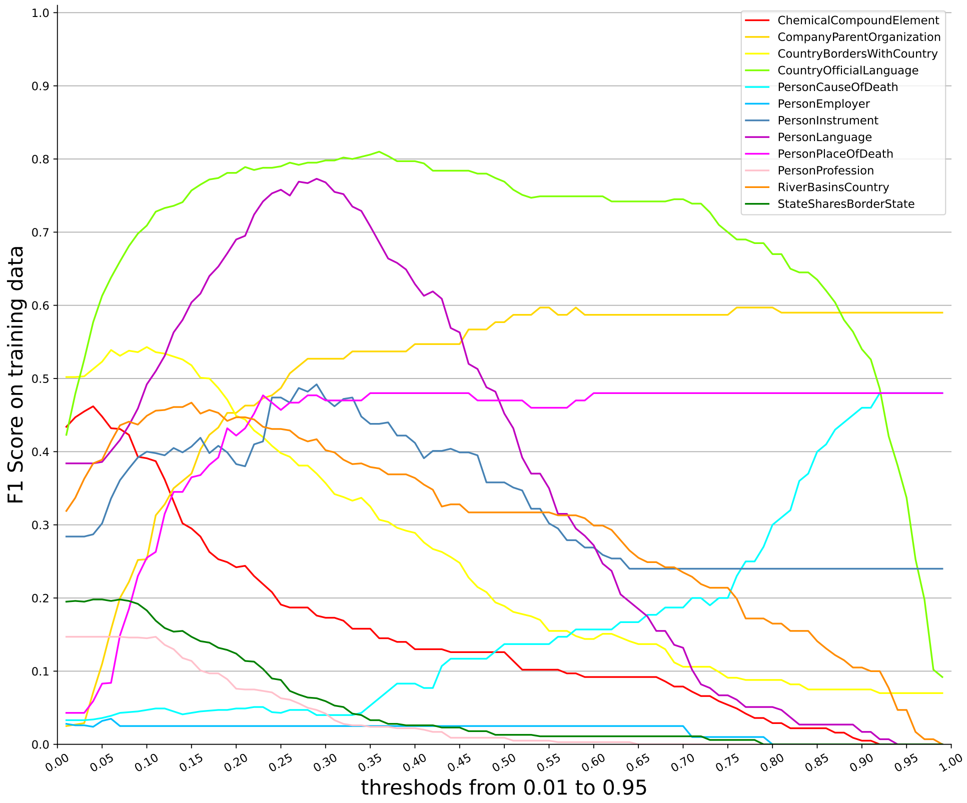

Figure 1: The effect of thresholds on the F1 for relations on training data

information extraction. In this paper, we have applied different prompting techniques to extract factual knowledge from pre-trained language models. We also designed various templates to generate diverse prompts to query specific pieces of relational knowledge. Experiments show that LMs are indeed reliable knowledge sources than initially indicated by previous results, but they are also quite sensitive to the way we query them. We have made significant success compared to the baseline method by generating more effective prompts, ensemble prompts and search different thresholds. It is promising to improve the accuracy of factual knowledge retrieval by prompt design strategies for each relation. However, how to create a prompt, how to select the language model, how to construct answer candidates, how to map answers to final outputs, and how to find an optimal configuration for downstream tasks is still an on-going exploration.

6. Acknowledgments

Thanks to Shuai Wang, an excellent software engineer from Amazon, he introduced several practical scripts for me to automate run the code, which significantly increased the efficiency of experiments. Furthermore, the experiment part of this research was made possible, in part, using the Data Science Research Infrastructure (DSRI) hosted at Maastricht University.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[3] 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, arXiv preprint arXiv:2107.13586 (2021).

[4] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, arXiv preprint arXiv:2012.15723 (2020).

[5] 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.

[6] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, arXiv preprint arXiv:2010.15980 (2020).

[7] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using bart, arXiv preprint arXiv:2106.01760 (2021).

[8] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, H. Wu, Unified structure generation for universal information extraction, arXiv preprint arXiv:2203.12277 (2022).

[9] A. Lazaridou, E. Gribovskaya, W. Stokowiec, N. Grigorev, Internet-augmented language models through few-shot prompting for open-domain question answering, arXiv preprint arXiv:2203.05115 (2022).

[10] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1499–1509.

[11] C. Carpineto, G. Romano, A survey of automatic query expansion in information retrieval, Acm Computing Surveys (CSUR) 44 (2012) 1–50.

Table 2
The final selected prompt(s) and corresponding threshold.

Relations	Prompts	Thresholds
ChemicalCompoundElement	s consists of m, which is an element.	0.04
CompanyParentOrganization	The parent organization of s is m company. s is part of m. The parent company of s is m company.	0.54
CountryBordersWithCountry	s shares border with m. s bordered by m. s adjacents to m.	0.1
CountryOfficialLanguage	The official language of s is m. The language spoken in s is m The language official in s is m.	0.36
PersonCauseOfDeath	s died due to m. s died of m. s died from m disease.	0.92
PersonEmployer	s joined m company. s is employed by m. s is the chairman of m.	0.06
PersonInstrument	The musician s plays m, which is an instrument. The musician s plays m. The musician s plays instrument of m.	0.29
PersonLanguage	s speaks in m. s writes language m. s second language is m.	0.30
PersonPlaceOfDeath	s died at m. s died in m. The death place of s at m. The death location of s at m.	0.95
PersonProfession	s is a m by profession. s worked as a m. s has the job of m. s employed as a m. s is a m, which is an occupation requiring special education. s received a specialized professional training and became a m.	0.01
RiverBasinsCountry	s river basins in m. the watershed in m is s.	0.15
StateSharesBorderState	s state next to m state. s state borders m state. s state bordered by m state. s state adjacent to m state.	0.04

s denotes $\{subject\_entity\}$, o denotes $\{mask\_token\}$.

Table 3
Performance of our system on test dataset.

Relations	Baseline			Ours		
	Precision	Recall	F1-score	Precision	Recall	F1-score
ChemicalCompoundElement	0.98	0.069	0.098	0.585	0.508	0.4528
CompanyParentOrganization	0.90	0.740	0.640	0.760	0.770	0.6800
CountryBordersWithCountry	0.98	0.1046	0.1187	0.575	0.6202	0.5473
CountryOfficialLanguage	0.98	0.7185	0.786	0.7307	0.7968	0.8253
PersonCauseOfDeath	0.86	0.500	0.36	0.98	0.500	0.500
PersonEmployer	0.98	0.020	0.020	0.016	0.060	0.0267
PersonInstrument	1.00	0.360	0.360	0.78	0.6072	0.5203
PersonLanguage	1.00	0.3757	0.428	0.9267	0.7487	0.7877
PersonPlaceOfDeath	0.98	0.500	0.48	0.98	0.500	0.480
PersonProfession	1.00	0.00	0.00	0.412	0.2698	0.3025
RiverBasinsCountry	0.96	0.404	0.429	0.6133	0.5379	0.5129
StateSharesBorderState	0.90	0.0057	0.010	0.2983	0.1962	0.2870
Average	0.96	0.3165	0.3108	0.6381	0.5096	**0.4935**