Expanding the Vocabulary of BERT for Knowledge Base Construction

Dong Yang[1], Xu Wang[1] and Remzi Celebi[1]

[1]Institute of Data Science, Department of Advanced Computing Sciences, Maastricht University, The Netherlands

Abstract

We present an approach to construct knowledge bases with Language Models. Our approach involves task-specific pre-training to improve the language model of predicting the masked object tokens and token re-coding to expand the vocabulary of the language model for higher-quality retrieval. Our approach achieves 32.26% F-1 score on the hidden test set of the challenge.[1]

1. Introduction

Knowledge bases have a profound impact across diverse domains, offering transformative benefits. They enhance information retrieval systems, leading to increased efficiency and accuracy, thereby enabling users to swiftly locate relevant data [1]. In the context of natural language processing, knowledge bases play a crucial role in elevating semantic comprehension and facilitating a range of language-related tasks [2]. Moreover, these knowledge bases actively promote data integration and interoperability, making substantial contributions to the advancement of initiatives such as the Semantic Web and Linked Data [3]. In this work, we present our approach for the LM-KBC challenge at ISWC 2023, which focuses on knowledge base construction for 21 relations. The task of the challenge involves predicting objects based on given subject-relation pairs. For example, given the subject-relation pair *<South Korea, CountryBordersCountry>*, the goal is to predict appropriate objects such as *Japan, People's Republic of China, North Korea* . In this challenge, each participant receives a set of subject-relation pairs and is tasked with identifying the appropriate objects for these pairs. Each subject-relation pair can be associated with zero, one, or multiple true objects, reflecting the complex nature of real-world scenarios.

We participate in **track 1** of the challenge, which limit the parameters of language model up to 1 billion. According to our results (Tabel 4), the generative model achieved poor performance. Thus we selected the BERT [4] model as encoder and performed Filled-Mask task to retrieve object candidates [5] . For each [mask] token, the language model independently assigns confidence score to all tokens within its vocabulary. However, the original filled-mask task was designed to select the best single candidate, rather than multiple top candidates. Furthermore, the target object entity may consist of multiple tokens, and for a given subject-relationship

[1]Our code and data are available at https://github.com/MaastrichtU-IDS/LMKBC-2023

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

dong.yang@maastrichtuniversity.nl (D. Yang); xu.wang@maastrichtuniversity.nl (X. Wang); remzi.celebi@maastrichtuniversity.n (R. Celebi)

South	Korea	shares	its	borders	with	the	[MASK]	[MASK]	[MASK]	[MASK]	[MASK]	[MASK]
							'Japan'					
							People	‘	s	Republic	of	China
							North	Korea				

(a) Problem of Predicting Candidates Consists of Different Number of Tokens

South	Korea	shares	its	borders	with	the	[MASK]
							'Japan'
							People's Republic of China
							North Korea

(b) Combining Candidates into One Token

Figure 1: Example of Object Consists of Multiple Tokens

pair, the number of potential objects can vary. The original model for filled-mask tasks is not inherently formulated to predict multiple objects comprised of numerous tokens. For example (Figure 1 (a)), the correct answers of given subject-relation pair *<South Korea, CountryBordersCountry>* are combinations of tokens, such as *('Japan'), ('People', "'", 's', 'Republic', 'of', 'China'), and ('North', 'Korea')*. However, extracting entities from the filled-mask task is not a straightforward process, as it presents an permutation problem due to the language model's independent prediction of each token.

To be able to predict multiple candidates with multiple tokens, we expand the vocabulary of language model (e.g. BERT), and the object comprising multiple tokens is treated as a distinct token. As shown in Figure 1 (b), our approach entails grouping token combinations into single tokens respectively. However, a drawback of this method is that the newly created tokens cannot leverage the information provided by the language model.

To address this challenge, we introduce a novel approach called Token Re-code (TR) that aims to provide an initial semantic vector for these entities. Thus we build a vocabulary for predicting candidates. Empirical experiments have been conducted to evaluate the effectiveness of TR, and the results demonstrate an obvious improvement in the f1 score. This indicates that leveraging the Token Re-code task enhances the model's ability to align newly defined entities with their constituent tokens, thereby providing a more accurate semantic representation.

We collect sentences from wiki according to our vocabulary. Experiments shows the task-specific pre–train is effective. We also categorised the entities and applied category-based filter for the object candidates.

The main contributions of this paper is:

- Proposing a novel method to expand the vocabulary of a language model (E.g. BERT) while preserving the semantic meaning of the newly added entities.

- Conducting experiments to verify the effectiveness of pre-train on raw text for knowledge base construction
- Category-base filter and adaptive thresholding to further enhance the performance.

We achieved 0.33 at validation set and 0.31 on hidden test set with bert-base-cased model, which is a relative light language model, with only 110 million parameters.

2. Related Work

2.1. Language Model

The Bidirectional Encoder Representations from Transformers (BERT) [4] model has transformed the field of natural language processing (NLP) since its introduction. BERT's primary contribution lies in its ability to generate contextualized word embeddings, capturing bidirectional context information and producing rich semantic representations. By pre-training on large-scale unlabeled text using a masked language modeling objective, BERT learns a deep representation of language structures, enabling it to capture complex linguistic patterns and relationships.

Besides, The XLNet and GPT (Generative Pre-trained Transformer) models are two other prominent advancements in the field of natural language processing (NLP). Both models have made use of transformers and self-supervised learning techniques, leading to the significant contributions to language understanding and generation tasks.

Gururangan [6] built separate pretrained models for specific domains with a universal language model ROBERTA on four domains (biomedical and computer science publications, news, and reviews and eight classification tasks (two in each domain). Their experiments showed that continued pre-training with additional corpus on the domain consistently improves performance on tasks from the target domain, in both high- and low-resource settings.

2.2. Knowledge Base Construction

Li [5] proposed a model, which is based on BERT-large-cased, to improve performance in the following three dimensions: (1) LM representation of masked object tokens;(2) entity generator; (3) candidate object selection. Language models are trained on a diverse corpus, thus show weaker performance on specific domain-related tasks. The author used additional triples to train the language model and therefore, leading significant improvement. The skills related to prompting can be categorized into four types: (1) incorporating type information to entities; (2) simplifying and condensing prompt (3) generating prompts by extracting relevant sentences from Wikipedia; (4) selecting different prompts for the same relation based on the type of entity. For example, a provincial-level administrative region is called 'state" in the US and "province" in the Netherlands. Besides, they remove pronouns and determiners from the candidates and find the optimal threshold of each object-relation pair and use original score of predictions rather than softmax.

3. Model

We first initialize the token embeddings and output embeddings of a preeminent pre-trained model, specifically "bert-base-cased" as referenced within this work. Subsequently, we execute filled-masked task for the aforementioned pre-trained model, employing a corpus sourced from the domain of Wikipedia. The selection of sentences for this task is contingent upon their incorporation of a quantified count of entities derived from our vocabulary.

Following this, we fine-tune the pre-trained model using the training set and apply the model to predict the object candidates with either the validation set or test set.

Besides, for the relations "PersonHasNumberOfChildren" and "SeriesHasNumberOfEpisodes", we check whether the resulting candidates for an object is a number. For all other relations, we select the best thresholds independently in validation set, and use the selected threshold in test set.

3.1. Fine-tune on knowledge base construction

Contemporary language models, exemplified by BERT [4], have undergone extensive training on a corpus of diverse textual data at a significant scale. This inherent capacity for comprehensiveness suggests that a process of "rekindling" the models' awareness of the specific categories of information they are expected to recall during fine-tuning could potentially yield performance enhancements. In congruence with this perspective, [5] have empirically demonstrated the utility of fine-tuning in augmenting the performance of language models in knowledge base construction.

The process of fine-tuning for the knowledge bases construction can be summarized as follows: given a subject-relation-object triple, this triple is transformed into a coherent sentence using a corresponding prompt template. Relevant tokens related to the object entity are hidden within the sentence. The subsequent task involves training BERT models using the masked sentence as input, aiming to effectively uncover the hidden tokens.

3.2. Vocabulary

We categorized the entity by their role in a tripe (subject or object).

To predict new objects, we create a task-specific vocabulary using entities from the training, validation, subject entities form test set, and our silver set. The table of the vocabulary is:

The BERT model is primarily designed for a "filled-mask" or "masked language modeling" task, where certain tokens in a sentence are masked, and the model is trained to predict the original tokens. The objective is to learn contextualized representations of words that take into account their surrounding context.

3.3. Token Recode

The BERT (Bidirectional Encoder Representations from Transformers) [4] model uses two fundamental types of embeddings: input embeddings and output embeddings, each of which serves a distinct yet interrelated role in the model's functioning.

Table 1
The entity type of relation

Relation	Subject Type	Object Type
BandHasMember	Band	Person
CityLocatedAtRiver	City	River
CompanyHasParentOrganisation	Company	Company
CompoundHasParts	Compound	Part
CountryBordersCountry	Country	Country
CountryHasOfficialLanguage	Country	Language
CountryHasStates	Country	State
FootballerPlaysPosition	Person	Position
PersonCauseOfDeath	Person	Cause
PersonHasAutobiography	Person	Autobiography
PersonHasEmployer	Person	Company
PersonHasNoblePrize	Person	Prize
PersonHasNumberOfChildren	Person	Number
PersonHasPlaceOfDeath	Person	City
PersonHasProfession	Person	Profession
PersonHasSpouse	Person	Person
PersonPlaysInstrument	Person	Instrument
PersonSpeaksLanguage	Person	Language
RiverBasinsCountry	River	Country
SeriesHasNumberOfEpisodes	Series	Number
StateBordersState	State	State

The input embeddings play a crucial role in representing the textual input in BERT by capturing the inherent semantic and contextual information of the input tokens. Specifically, BERT utilizes WordPiece embeddings, which break down words into sub-word units (sub-tokens). These sub-word embeddings are then combined with positional embeddings to encode both the content and the position of the tokens within the input sequence. The input embeddings go through a series of transformations as they pass through BERT's layers. Initially, these embeddings are fed into the model's self-attention mechanism, which enables the model to capture contextual relationships between tokens in both directions (left-to-right and right-to-left) in the input sequence. This bidirectional context is a significant departure from previous models that relied solely on left-to-right or right-to-left information flow.

The output embeddings refer to the representations of the tokens that are obtained after the input embeddings have been processed through BERT's layers. These output embeddings encapsulate the model's learned understanding of the input text's semantics and context. Output embeddings can be utilized for various downstream tasks, such as text classification, named entity recognition, question answering, and more.

Given an input sequence of tokens $X = (x_1, x_2, ..., x_n)$, where each x_i represents a token (word or sub-word) in the sequence. The input tokens are first transformed into embeddings $E = (e_1, e_2, ..., e_n)$, where each e_i is the embedding representation of token x_i. These embeddings

Table 2
The count of entity types within our vocabulary

Entity Type	Number
Company	1654
Country	248
Language	252
Number	1063
City	111
Profession	233
Instrument	58
State	5470
Person	2287
River	1999
Position	78
Cause	78
Autobiography	375
Part	407
Prize	7
Total	14320

are then processed through multiple layers of Transformer architecture.

$$P(x_i|e_1, e_2, ..., e_n) = T(E) \times O \tag{1}$$

Where $P(x_i|e_1, e_2, ..., e_n)$ is the predicted probability distribution over the vocabulary for the i-th position, conditioned on the embedding of all tokens in the sequence. The E refers to the token embeddings of the input tokens X. The T refers stacked transformers layers. $O \in R^{l \times v}$, where l is the length of the width of last hidden layer of the T, v is the number of the vocabulary.

We introduce modifications to the token embedding and output embedding components of the BERT model, enabling the generation of embeddings for newly introduced words based on their constituent tokens. For instance, consider the entity 'United States of America', which is segmented into individual tokens as ['United', 'States', 'of', 'America']. Consequently, the token and output embeddings for the phrase 'United States of America' are computed as the average of the respective embeddings for the tokens ['United', 'States', 'of', 'America'].

We count how often each token (represented as C) appears in a sentence sourced from a knowledge base like Wikipedia. We suggest that tokens that appear more frequently hold less importance. Thus, we calculate weights for the tokens based on the reciprocal of their frequency counts. We then normalize these weights to create a standardized representation. The equation is as following:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{C_i} \tag{2}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2} \tag{3}$$

$$W_i = \frac{(T_i - \mu)}{\sigma} \tag{4}$$

In a more general context, we denote the newly introduced word as W, with its associated tokens represented as $S = (s_1, s_2, s_3, \dots, s_n)$. The initial token embeddings for these tokens are denoted as $t = (t_1, t_2, t_3, \dots, t_i, \dots, t_n)$, while the original output embeddings are denoted as $o = (o_1, o_2, o_3, \dots, o_i, \dots, o_n)$. Subsequently, the new token embedding for word W is denoted as T_{new}, and the new output embedding is denoted as O_{new}. Then we obtain the token embedding and output embedding of a word by the original token embedding t and output embedding o of its corresponding tokens:

$$T_{new} = \sum_{i=1}^{n} W_i \cdot t_i \tag{5}$$

$$O_{new} = \sum_{i=1}^{n} W_i \cdot o_i \tag{6}$$

4. Pre-training on Wikipedia

We generate the embedding of a word of BERT model by deriving the embedding of its constituent tokens. Furthermore, we pre-train the model by conducting filled-mask task on our collected Wikipedia corpus. The sentence in our corpus is selected based on the criterion that the sentences encompass the entities listed in our vocabulary. Table 3 indicates the number of sentence that include specific entity type. Please note that a sentence can contain multiple entities, resulting in the cumulative count of sentences for each entity type being greater than the actual overall sentence count.

5. Experimental setup

We use transformers on pytorch to build our system, conducting experiments on laptop 3080ti. For pre-training on wiki sentence task, the learning-rate is set to $2e^{-5}$ and epoch numbers is 20 . For fine-tune task, the learning-rate is set to $2e^{-5}$ and the number of epoch is 20. The code is available at https://github.com/MaastrichtU-IDS/LMKBC-2023

6. Results

Table 4 summarizes the results of our experiments. Our Token Recode methods successfully help extract correct object entities. Pre-training on Wikipedia improves constructing knowledge bases. Finally, combining the token-recode method and pre-training achieves the best result on validation set.

Table 5 shows the final results for our system on the challenge validation set. This final results is acquired under the following setup: we initialize the token embedding and output embedding of the BERT-base-cased; We perfomed filled-mask task for mentioned model on Wikipedia sentences collected for this task. We applied the thresholds found on Valid set.

Table 3
The quantity of sentences for each entity type

Entity Type	Quantity of Sentences
Person	22114
Country	10845
Series	3665
State	36076
Company	12981
Band	5246
River	9770
Autobiography	879
City	10015
Compound	408
Profession	5062
Part	649
Position	444
Cause	1041
Language	3732
Instrument	932
Prize	576
Total	51496

Table 4
Results on validation set with incrementally applying various techniques

Method	Precision	Recall	F-1
$baseline_{bert-base-cased}$	0.131	0.474	0.112
$baseline_{facebook/opt-1.3b}$	0.073	0.101	0.039
$fine-tune_{adaptive-threshold}$	0.28	0.31	0.26
category-base filter	0.4155	0.3495	0.2933
token-recode	0.4576	0.3555	0.3035
pretrain on sentences from wiki	0.3866	0.4235	0.3401

7. Conclusions

Our research has focused on improving the way we build knowledge bases by using small Language Models. We followed the guidelines set in **track 1**, where BERT is used as the main language model. In a focused manner, our investigations have encompassed the expansion of BERT's lexical repository. We enhance the representations of language model (i.e. BERT) via pre-training on task-specific corpus and fine-tuning on tran set. Besides, we adopt category-based filters and adaptive threshold. This holistic exploration has yielded substantial advancements beyond the established baseline methodology.

While there are indeed several challenges associated with extracting factual statements from language models, we place particular emphasis on the importance of the following areas for future research: the development of an efficient algorithm for token recoding and token recoding in generative models.

Table 5
The results of baseline and ours system on validation set

Relation	Baseline			Ours		
	Precision	Recall	F-1	Precision	Recall	F-1
BandHasMember	0.000	0.960	0.000	0.1234	0.0354	0.0255
CityLocatedAtRiver	0.018	0.145	0.020	0.2620	0.1750	0.1283
CompanyHasParentOrganisation	0.060	0.640	0.053	0.7900	0.6200	0.6100
CompoundHasParts	0.164	0.475	0.179	0.8149	0.7947	0.7795
CountryBordersCountry	0.372	0.709	0.443	0.5319	0.5339	0.4774
CountryHasOfficialLanguage	0.760	0.685	0.668	0.8103	0.7246	0.7145
CountryHasStates	0.002	0.076	0.005	0.2272	0.0690	0.0885
FootballerPlaysPosition	0.413	0.143	0.206	0.1534	0.7417	0.2464
PersonCauseOfDeath	0.040	0.012	0.019	0.7100	0.6800	0.6800
PersonHasAutobiography	0.000	0.950	0.000	0.0000	0.0000	0.0000
PersonHasEmployer	0.000	0.970	0.000	0.2314	0.0583	0.0394
PersonHasNoblePrize	0.000	0.000	0.000	0.6983	0.6900	0.6050
PersonHasNumberOfChildren	0.000	0.000	0.000	0.2702	0.7800	0.3973
PersonHasPlaceOfDeath	0.091	0.641	0.088	0.6162	0.5354	0.5152
PersonHasProfession	0.000	0.700	0.000	0.1028	0.2068	0.1228
PersonHasSpouse	0.000	0.890	0.000	0.0000	0.0000	0.0000
PersonPlaysInstrument	0.000	0.110	0.000	0.3123	0.5058	0.3580
PersonSpeaksLanguage	0.383	0.310	0.316	0.6760	0.6783	0.6306
RiverBasinsCountry	0.423	0.402	0.345	0.5758	0.4667	0.4731
SeriesHasNumberOfEpisodes	0.000	1.000	0.000	0.0887	0.4500	0.1472
StateBordersState	0.016	0.132	0.016	0.1243	0.1489	0.1031
Average	0.131	0.474	0.112	0.3866	0.4235	0.3401

Acknowledgments

The authors thank the challenge organizers for their timely and helpful response to inquiries, and the reviewers for their valuable comments. This work is supported by China Scholarship Council (202207010004).

References

[1] H. D. Nguyen, T.-V. Tran, X.-T. Pham, A. T. Huynh, V. T. Pham, D. Nguyen, Design intelligent educational chatbot for information retrieval based on integrated knowledge bases, IAENG International Journal of Computer Science 49 (2022) 531–541.

[2] T. Zhang, C. Wang, N. Hu, M. Qiu, C. Tang, X. He, J. Huang, DKPLM: Decomposable Knowledge-Enhanced Pre-trained Language Model for Natural Language Understanding, Proceedings of the AAAI Conference on Artificial Intelligence 36 (2022) 11703–11711. URL: https://ojs.aaai.org/index.php/AAAI/article/view/21425. doi:10.1609/aaai.v36i10.21425, number: 10.

[3] S. Bouaicha, W. Ghemmaz, A Semantic Interoperability Approach for Heterogeneous Meteorology Big IoT Data, in: M. R. Laouar, V. E. Balas, B. Lejdel, S. Eom, M. A. Boudia

(Eds.), 12th International Conference on Information Systems and Advanced Technologies "ICISAT 2022", Lecture Notes in Networks and Systems, Springer International Publishing, Cham, 2023, pp. 214–225. doi:10.1007/978-3-031-25344-7_20.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[5] T. Li, W. Huang, N. Papasarantopoulos, P. Vougiouklis, J. Z. Pan, Task-specific Pre-training and Prompt Decomposition for Knowledge Graph Population with Language Models, 2022. URL: http://arxiv.org/abs/2208.12539, arXiv:2208.12539 [cs].

[6] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't Stop Pretraining: Adapt Language Models to Domains and Tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8342–8360. URL: https://aclanthology.org/2020.acl-main.740. doi:10.18653/v1/2020.acl-main.740.