BERT with context information encoding from knowledge graphs

Muhan Li, David Demeter
muhanli2022@u.northwestern.edu

Abstract

We introduce a method to extend the vocabulary encoding of BERT with context encoding containing rich information of a input token, in a given sequence of text. The context encoding is output by another BERT model, named as CTX-BERT, dedicated to infer relations to entities of the specified token in its context. To simplify the model, we combine there objectives: entity detection, entity encoding and relation recovery into one by requiring CTX-BERT to recover the relation triples as as a textual sequence when given a context sequence with the target token masked. Experimental results demonstrate that CTX-BERT could enhance the performance of the second BERT on question answering tasks.

1 Introduction

Language models based on the transformer architecture(Vaswani et al., 2017) and using unsupervised learning objectives for pre-training has achieved promising results on various NLP tasks such as question answering, text classification, sentiment analysis, etc. Many famous publicly available models including BERT (Devlin et al., 2018), GPT (Radford et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019) are pre-trained using large-scale generic corpora such as Wikipedia with millions of entries. However, in order to apply them to domain specific tasks, such as question answering on biomedical datasets (Gu et al., 2020), the performance of a generic model could be less than idea even after fine-tuning, due to the lack of knowledge on domain specific words and their distributions. Many methods such as extending the vocabulary of the model (Tai et al., 2020), fine-tune an ensemble of generic and domain-specific pre-trained models (Gu et al., 2020), incorporate domain related knowledge graph encodings (Yan et al., 2020) (Liu et al., 2020) (Ostendorff et al., 2019) are proposed to enhance the performance of a generic model on specific tasks related to one or more domains.

Methods which use a knowledge graph (KG) are preferred over others, due to its structural representation by formulating relations as a triple with a head entity, a tail entity, and a relation connecting the head and the tail. Such a simple form of knowledge allows many already existing KGs designed for symbolic reasoning to be utilized, and make the model a domain expert with relatively low training cost, when compared to redo the pre-training process on a domain specific corpora. The form is also generalizable to less constrained common-sense knowledge graphs (Ilievski et al., 2020) where the head and tail entity are often represented as textual sequences, thereby are not atomic and unique since multiple descriptions with the same meaning could exist.

There are several challenges lying in integrating knowledge graphs with currently existing pre-trained language models: **(1) Difficulty in architecture incorporation**: Currently existing graph encoding methods such as the graph neural networks (Zhou et al., 2020) are not able to encode graph structure along with textual sequences. **(2) Lack of annotation** For many textual corpora we lack annotation of entities in text and thus can not search for their relations in a knowledge graph. **(3) Heterogeneity in embedding**: Entities are unique and atomic special textual sequences, if they are encoded separately, then there needs to be an additional projection relationship between text and entity embedding.

In this paper, we propose a simple architecture to mix text encoding with entity relation encoding, the main contributions are summarized as follows:

- We propose CTX-BERT, which is able to recover the relation triples of a masked token

in a given context, this combines three objectives: entity detection, entity encoding, relation encoding as one and greatly simplifies the artitecture.

- We test CTX-BERT on multiple token embedding extension schemes, and relation encoding schemes, demonstrating that it could greatly improve the performance of traditional pre-trained models such as BERT on domain-specific tasks as well as open-domain tasks.

2 Related work

While there is an abundance of studies on optimizing pre-training procedure such as (Liu et al., 2019), (Sanh et al., 2019), (Lan et al., 2019), fusing pre-trained models with knowledge graphs gains relatively low attention and has a much smaller flock of works. In (Ostendorff et al., 2019), they use a 2-layer MLP with the concatenation of the output from BERT and graph meta data, to combine textual information with graph information. In the study of Jun et al (Yan et al., 2020), such a combination method using MLP is also used. KI-BERT proposed in (Faldu et al., 2021) use a special token embedding transformed from word embeddings along with knowledge graph embeddings from ConceptNet for atomic entity representation with text. In (Liu et al., 2020), K-BERT encode a sentence tree comprised of a trunk representing the input textual sequence and branches for triples in knowledge graphs, they flatten this structure using a visible matrix which uniquely encodes the topology, however, both KI-BERT and K-BERT requires manual injection of knowledge triples and therefore entity detection and normalization must be performed in advance.

Other works related to both knowledge graphs and pre-trained language models include entity normalization (Ji et al., 2020), entity recognition (Souza et al., 2019), relation extraction from text (Xue et al., 2019), graph completion for partial triples (Zhao et al., 2020) (Yao et al., 2019), etc. In most of these works, relation triples are flattened into a textual sequence, and the mask token in masked language modeling, also known as one of the pre-training objectives in BERT, is then used to shadow the entity or relation that needs to be predicted. Some works such as (Xue et al., 2019) use the MASK matrix or the attention matrix in BERT to direct attention to specific parts of the input triple while ignoring others and gain context-sensitive embeddings.

In this work, we mix previously mentioned ideas into two objectives: reconstructing a simple knowledge graph related to a token in context, and enforce relationship between tokens, this formulation allows us to obtain a context-sensitive embedding while achieving entity recognition and relation extraction using one model, thus greatly reduces the complexity.

3 Model

In this section, we will outline the structure and implementation of our model, as well as the training objectives..

3.1 Notation

Let input sentence $s = [w_0, w_1, \ldots, w_{n-1}]$ be the context sequence, with $w_0, \cdots w_{n-1}$ being tokens. Let an entity referenced in this sentence be $e = [w_i, w_{i+1}, \ldots, w_j], 0 \leq i < j \leq n-1$. Let the sub-knowledge graph $G_e = (V_e, E_e) \subseteq G$ be a subgraph of the complete knowledge graph G, with max-depth D and root node $v^* = e$. Here, the max-depth is defined as $D = \max_{v \in V_e, v \neq v^*} d(v, v^*)$ with d being the distance function on graph.

3.2 CTX-BERT

The architecture of CTX-BERT is shown in Figure 1, for each token $w_k \in s$, the first input of CTX-BERT is the padded context window c around w_k with fixed size l:

$$c = [c_0, c_1, \ldots, c_{l-1}] \tag{1}$$

$$\forall c_m \in c, c_m = \begin{cases} w_{k+m-\frac{l-1}{2}}, & \text{if in } [0, n-1] \\ [\text{PAD}], & \text{else} \end{cases} \tag{2}$$

Next, for triples representing each edge in graph G_e, they are sorted in decreasing order by their depth, and flattened as a relation sequence r, note the bracket here represents concatenation. For each relation triple, a [PAD] token is inserted behind as a separator. We keep adding relation triples in sequential order until it exceeds maximum input length allowed by BERT, then the remaining space are filled with [PAD] tokens.

$$r = [v_0, e_0, v_0', [\text{PAD}], \ldots v_i, e_i, v_i', [\text{PAD}]] \tag{3}$$

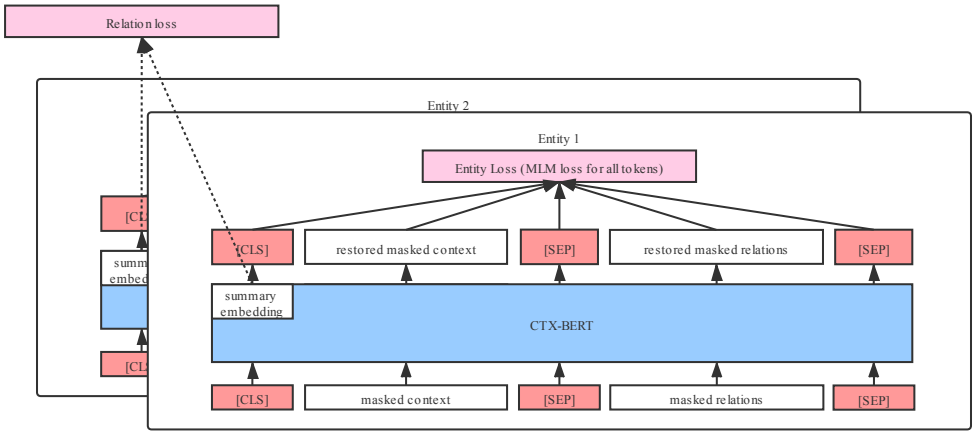

Figure 1: Structure and training objectives of CTX-BERT, training objectives are colored in pink.

The eventual input is constructed as the concatenation of the context sequence c and relation sequence r:

$$input = [[\text{CLS}], c, [\text{SEP}], r, [\text{SEP}]] \quad (4)$$

3.3 Training CTX-BERT

Two training objectives are used to train CTX-BERT, in actual experiments we fine tune pre-trained BERT models on these objectives rather than training BERT from the start.

3.3.1 Entity encoding objective

The entity encoding objective focus on letting BERT learn to recover the target entity name, when given a context where the target entity is entirely or partially masked. Since an entity may be referenced in a context with multiple tokens, and CTX-BERT is constrained to recover one masked token at a time, the following relation is added to the front of r:

$$r_{part} = [(\text{token}), \text{is a sub part of}, \ e] \quad (5)$$

Where (token) is one of all reference tokens. Then, for all tokens in r, the entity reference e is masked if it only has one token, otherwise (token) is masked instead. This setup ensures the model never sees the target masked token and must infer it from the given context c and relation r.

3.3.2 Relation encoding objective

We wish to also recover relation information between two encoded entities e_p and e_q, so that when r is partially observed or empty in the extreme case, the relation between these two entities could still be reconstructed from their context sequence c_p and c_q, implying that CTX-BERT is not just learning the entity name but also its true meaning and relations with other entities.

The last hidden state output corresponding to the [CLS] token is commonly used to perform classification tasks. Denoting the [CLS] hidden state output as o_p and o_q for two entities, two methods are experimented to recover the relation information.

Concatenation This method concatenates o_p and o_q and pass result through a MLP network, the relation is then the output of softmax:

$$r_{pq} = softmax(W^T[o_p, o_q] + b) \quad (6)$$

Subtraction Translational distance models are commonly used in many relational data embedding methods, where relations are encoded as translation vectors between two entity points, typical examples include TransE(Bordes et al., 2013) and TransD (Ji et al., 2015). Although this method is shown to be flawed when dealing with reflective/one-to-many/many-to-one/many-to-many relations (Nur et al., 2019), it could still serve as a baseline with interpretability. We generate a random matrix $M \in \mathbb{R}^{dim(o)\times|R|}$ with orthonormal columns, and each column represents a type of relation. The relation is then:

$$r_{pq} = softmax(|M^T(o_p - o_q)|) \quad (7)$$

We use the absolute value to ignore direction of each relation, the direction information could be later recovered by the sign function $sign(M^T(o_p - o_q))$.

3.4 Extending embeddings with CTX-BERT

The embedding output of CTX-BERT could then be utilized to enhance the performance of other language models using the same byte pair encoding (Sennrich et al., 2015), such as BERT itself and its descendents. We refer to this model as EXT-BERT. Three types of extending methods are used in experiments, let w be the original embedding of some token, and o be the output embedding from CTX-BERT.

Fixed ratio

$$w' = \alpha w + (1 - \alpha)o \tag{8}$$

With $alpha$ being a constant value in $(0, 1)$

Learnable ratio

$$w' = Aw + (1 - A)o \tag{9}$$

With A being a vector of the same size as embeddings. Compared to exBERT (Tai et al., 2020), we mix by dimension instead of mix by token.

Replace

$$w' = o \tag{10}$$

Replacing is actually a special case of the fixed-ratio method when $\alpha = 0$

4 Experiments

In this section we will describe details of experiments.

Datasets The CTX-BERT model is trained on the Kensho Derived Wikimedia Dataset (KDWD) (Altay, 2020) produced by cross linking text data in Wikipedia with relational data from Wikidata. EXT-BERT is trained on SQuAD 2.0 .

Settings We first train EXT-BERT on KDWD, in each iteration a batch for entity encoding training and a batch for relation encoding training are sampled together and loss on both objectives is summed as the total loss. For dataset the context sequence length is 200 and remaining are reserved for the relation sequence. The relation graph depth is limited to 1 and relation size is limited to 200, with other relations grouped as "unknown relation". For model, we choose a pre-trained BERT model with 12 layers, 768 hidden dimensions, 12 heads, and 110 Million parameters. Other configurations are: batch size=32, learning rate=5e-5, l2-regularization=1e-5, no drop out is used. For EXT-BERT the same training configurations are used. When CTX-BERT encodes the input tokens of EXT-BERT, the relation sequence is filled with [MASK] as no relation is known in advance.

Table 1: Result of EXT-BERT compared to BERT-BASE trained on SQuAD 2.0

System	Dev		Test	
	EM	F1	EM	F1
BERT-Base	74.96	77.60	78.03	81.57
EXT-BERT	76.48	78.85	80.21	83.32

EXT-BERT result The result of training EXT-BERT with extension embeddings from CTX-BERT are shown in Table 1, our model gains more than 2 points on the EM score and nearly 2 points on the F1 score when compared to the vanilla BERT with no modifications.

CTX-BERT output example We tested the encoding ability of CTX-BERT by manually providing input examples with entity token masked, then entity token may have multiple meanings and CTX-BERT must perform disambiguation using the context.

- Input: context="Apple, based in Cupertino, CA, is one of the most valuable companies in the world. It produces popular digital gadgets.", the first token "Apple" is masked.
- Output relation="apple ⟨is a sub token of ⟩apple inc. [PAD] apple inc. ⟨has works in the collection ⟩design museum gent ..."

5 Conclusions

In this work, we proposed a novel method named CTX-BERT for combined entity detection, entity and relation encoding from a given sequence of context. CTX-BERT is able to enrich information contained in tokens, and demonstrates promising results on enhancing the performance of the second model with extended vocabulary on question answering. CTX-BERT is also able to complete several entity related tasks while being relatively simple in architecture compared to previous method.

Some future directions of this study include testing our model on other tasks of NLP such as classification and sentiment analysis, and using bootstrapping to extend the model on less cross-linked and annotated datasets in contrast to KDWD.

Acknowledgments

We would like to thank David Demeter for providing many useful inspirations used in this study and support.

References

Gabriel Altay. 2020. Announcing the kensho derived wikimedia dataset.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Keyur Faldu, Amit P. Sheth, Prashant Kikani, and Hemang Akabari. 2021. KI-BERT: infusing knowledge context for better language and domain understanding. *CoRR*, abs/2104.08145.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *CoRR*, abs/2007.15779.

Filip Ilievski, Pedro A. Szekely, and Bin Zhang. 2020. CSKG: the commonsense knowledge graph. *CoRR*, abs/2012.11490.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, Beijing, China. Association for Computational Linguistics.

Zongcheng Ji, Qiang Wei, and Hua Xu. 2020. Bert-based ranking for biomedical entity normalization. *AMIA Summits on Translational Science Proceedings*, 2020:269.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nasheen Nur, Noseong Park, Kookjin Lee, Hyunjoong Kang, and Soonhyeon Kwon. 2019. Two problems in knowledge graph embedding: Non-exclusive relation categories and zero gradients. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1181–1186.

Malte Ostendorff, Peter Bourgonje, Maria Berger, Julian Moreno-Schneider, Georg Rehm, and Bela Gipp. 2019. Enriching bert with knowledge graph embeddings for document classification. *arXiv preprint arXiv:1909.08402*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Wen Tai, H. T. Kung, Xin Dong, Marcus Comiter, and Chang-Fu Kuo. 2020. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1433–1439, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Kui Xue, Yangming Zhou, Zhiyuan Ma, Tong Ruan, Huanhuan Zhang, and Ping He. 2019. Fine-tuning bert for joint entity and relation extraction in chinese medical text. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 892–897. IEEE.

Jun Yan, Mrigank Raman, Tianyu Zhang, Ryan A. Rossi, Handong Zhao, Sungchul Kim, Nedim Lipka, and Xiang Ren. 2020. Learning contextualized knowledge structures for commonsense reasoning. *CoRR*, abs/2010.12873.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Zhenjie Zhao, Evangelos Papalexakis, and Xiaojuan Ma. 2020. Learning Physical Common Sense as Knowledge Graph Completion via BERT Data Augmentation and Constrained Tucker Factorization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3293–3298, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.