Enriching Language Models Representations via Knowledge Graphs Regularisation

Matteo Medioli, Andrea Valenti and Davide Bacciu *

University of Pisa - Department of Computer Science
Largo B. Pontecorvo, 3 56127 Pisa - Italy

Abstract. In this paper, we propose a novel method for augmenting the representations learned by Transformer-based language models with the symbolic information contained into knowledge graphs. We first compute the node embeddings of a knowledge graph via a deep graph network. We then add a new regularisation term to the loss of BERT that encourages the learned word embeddings to be similar to the node embeddings. We test our method on the challenging WordNet and Freebase knowledge graphs. The results show that the regularised embeddings perform better than standard embeddings on the chosen probing tasks.

1 Introduction

In recent years, Transformer-based language models (LM) have become the defacto standard for solving a great variety of natural language processing tasks [1]. These models are, in general, able to learn rich word representations by leveraging massive amounts of unsupervised text corpora [2]. While learning these word-level embeddings, these LMs are able to take into account the *context* in which a particular word appear in order to make its representation more expressive.

However, sometimes context is not enough. In particular, the context of a word is very unlikely to explicitly provide many different kinds of potentially relevant information, such as, for example, linguistic information (e.g. “apples” is a plural noun) or general trivia about the world (e.g. “Galileo Galilei” was born in “Pisa”). It is therefore reasonable to expect that finding a way to inject this information into the learned word representation of LMs can be beneficial for possible downstream tasks. Nowadays, many of these types of information are freely available in public Knowledge Graphs (KG) such as WordNet [3] and Freebase [4].

In this paper, we enrich the word representations learned by a Transformers-based LM [2] with the knowledge coming from the aforementioned KGs. We do this by first learning a node-level embedding of the KG. We then use the learned node embeddings to regularise the word-level embeddings learned by the LM, with the goal to inject the information of the KG into the LM. We run a series of probing tasks on the resulting representations, to inspect their expressive power and the actual amount of information contained in them[1].

*The work has been partially supported by the EU H2020 TAILOR project (n.952215).

[1] The code of the experiments is available at `https://github.com/matteomedioli/BERT-KG`

2 Related Works

The relationship between node-embeddings and word-embeddings is suggested by the different applications of Deep Graph Neaural Networks [5] in language modeling, text-classification or combined with Transformer architectures [6].

The work proposed in [7] aim to derive LM's word-embeddings only from WN18RR [3] employing KBGAT [8]. Another approach proposes a model for context-aware paper citation recommendation task using paper citation graphs [9]. VGCN-BERT [10] combines the capability of BERT with GCN for text classification, exploiting self-attention mechanism during BERT fine-tuning to combine graph-embeddings and sentence-embeddings.

The task of context transfer between low-dimensional embeddings can be implemented as a linear regression problem. ALC embeddings [11] demonstrate how embedding regression allows context information to be inferred in a word representation, applying a single linear transformation. DMNE model [12] coordinates multiple graph neural networks with a co-regularised loss function to manipulate cross-graph relationships, applying linear regression to regularise different node-embeddings' latent spaces.

3 Enriching Language Models Representations

The proposed approach is articulated in three steps. First, we need to learn node-embedding representation of a given KG. Then, we have to find a way to match each word in the text corpus with the appropriate node embedding. Finally, we can train our LM using the novel regularisation term coming from the node embeddings.

Learning Knowledge Graph Embeddings. In order to learn the node embeddings of a KG, we use the KBGAT model described in [8]. The initial vector representation $\mathbf{h}_v^0$ is processed through several attention layers in order to aggregated the information coming from the neighbouring nodes. The representation of node v at layer ℓ is computed as

$$\mathbf{h}_v^\ell = \sigma\left(\sum_{u \in \mathcal{N}_v} \beta_{uv}^\ell \mathbf{W}^\ell \left[\mathbf{h}_u^{\ell-1} | \mathbf{h}_v^{\ell-1} | \mathbf{a}_{uv}^{\ell-1}\right]\right) \tag{1}$$

where $\mathcal{N}_v$ is the set of neighbours of v, $\mathbf{h}_v^{\ell-1}$, $\mathbf{h}_u^{\ell-1}$, $\mathbf{a}_{uv}^{\ell-1}$ are respectively the embedding of node v, the embedding of neighbouring node u and the embedding of edge between u and v at the previous layer, $\mathbf{W}^\ell$ is the trainable parameter matrix of the KBGAT layer, and β_{uv}^ℓ is the attention score between u and v.

The node embeddings $\mathbf{H}^\ell$ and edge embeddings $\mathbf{G}^\ell$ are obtained by concatenating the embeddings computed with Eq. 1:

$$\mathbf{H}^\ell = \left[\mathbf{h}_1^\ell | \dots | \mathbf{h}_{|\mathcal{V}|}^\ell\right], \quad \mathbf{G}^\ell = \mathbf{W}_G^\ell \left[\mathbf{a}_1^\ell | \dots | \mathbf{a}_{|\mathcal{E}|}^\ell\right], \tag{2}$$

where $\mathbf{W}_G^\ell$ are learnable model's parameters. This process is repeated for each graph attention layer. The output embeddings are in turn used to learn the set of triples that describe the KG. The model's objective is a pairwise margin loss over the set of negative and positive triples:

$$\mathcal{L}_{KBGAT} = \sum_{(\mathbf{h},\mathbf{r},\mathbf{t})\in\mathcal{T}} \sum_{(\mathbf{h}',\mathbf{r}',\mathbf{t}')\in\mathcal{T}'} \max\left(0, \tau + \|\mathbf{h}+\mathbf{r}-\mathbf{t}\| - \|\mathbf{h}'+\mathbf{r}'-\mathbf{t}'\|\right) \quad (3)$$

where $\mathcal{T}$ and $\mathcal{T}'$ are the set of positive and negative triples, respectively, composed by embeddings taken from the output matrices $\mathbf{H}^L$ and $\mathbf{G}^L$. τ is the margin parameter. The negative triples are constructed by randomly replacing either the head or tail entity of a positive triple.

Word Sense Disambiguation. One of the problem of matching a word-level embeddings in a text with node-level embeddings in a KG is that the same word can refer to multiple entities of the KG. It is therefore important to have a way to identify the correct meaning of the word, in order to match it with the correct node in the KG. To address this issue, given a word w in the input sentence, we first compute a *candidate set* of nodes $\mathcal{S}_w$. As proposed in [13], we then define an *injective mapping function* ϕ:

$$\phi(w, \mathbf{w}, \mathcal{S}_w, \mathcal{S}_\mathbf{w}) = \arg\max_{v\in\mathcal{S}_w} \sum_{\mathcal{S}\in\mathcal{S}_\mathbf{w}} \max_{u\in\mathcal{S}} \psi\left(v, u\right). \quad (4)$$

In the above equation, $\mathbf{w}$ is the set of *context words* (i.e. the words that appear in the same input sentence of w_i), and $\mathcal{S}_\mathbf{w}$ denotes all the candidate sets of the context words. For each word $w \in \mathbf{w}$, the function ϕ selects the optimal candidate node from the set $\mathcal{S}_w$ which maximises the node similarity function $\psi : \mathcal{V}\times\mathcal{V} \to \mathbb{R}$ with respect to all other candidate nodes in the candidate sets $\mathcal{S}_\mathbf{w}$ of the context words. For a given word w, the computed optimal candidate node represents the correct meaning given the input sentence $\mathbf{w}$. As node similarity function, we chose the Wu-Palmer similarity [14].

Language Model Regularization. After having learned the node embeddings of a KG and having identified a matching function between those node embeddings and the corresponding word embeddings in the text corpus, we are finally ready to "inject" the knowledge information contained in the KG nodes into the LM. The training objective of the LM is modified as follows:

$$\mathcal{L}_{\mathrm{MLM-KG}}\left(\mathbf{w}, \mathbf{H}^L, \Pi, \theta\right) = \frac{1}{K}\sum_{\pi\in\Pi} \log p\left(\mathbf{w}_\pi \mid \mathbf{w}_{\neg\Pi}, \theta\right) + \lambda\|\mathbf{H}^L - \mathbf{A}\mathbf{w}\|_2^2 \quad (5)$$

where $\mathbf{w}$ is the input sentence, Π is the set of K_Π indices of the masked words in the sentence, θ are the model's parameters, λ is the regularisation hyperparameter and $\mathbf{A}$ is a learnable matrix. The new loss takes in input the embedding matrix $\mathbf{H}^L$ computed over the KG in Eq. 2 and use it to regularise the

Table 1: Results of the regularised LM on the MEN probing task. MSE stands for mean squared error, ρ is the Spearsman's correlation.

Model	MSE	R^2	ρ
BERT-BASELINE	0.039	0.305	0.601
BERT-KG WN18RR	**0.036**	**0.362**	**0.652**
BERT-KG FB15k-237	0.039	0.311	0.614

unmasked word tokens of the input sentence. This additional term encourages the word embeddings learned by the LM to be similar (up to a linear transformation) to the ones produced by the graph neural network.

4 Training Setting

We assess our approach by considering two different KGs. WN18RR [3] is a subset of WordNet containing more than 90k relational triples with 40,943 entities and 11 relation types. In WN18RR, the node entities represents individual words while the edge model linguistic semantic relationships between those words. The second KG is FB15K-237 [4], containing triples and textual mentions of freebase entity pairs. Each triple models a real fact about the world. It contains more than 590k triples with 4951 entitities and 1345 relation types.

We use the same model for the two KGs, consisting in 2 KBGAT layers with dropout rate of $\rho = 0.3$ to prevent overfitting. Initial embedding matrix $\mathbf{H}^0$ is initialised with TransE [15] pre-trained embedding of dimension 50, while the model's output dimension is set to 200. Training is performed using the Adam optimizer, with learning rate $lr = 1 \times 10^{-3}$ and weight decay $lr = 5 \times 10^{-6}$. We use an exponential learning rate decay schedule, halving the learning every 500 epochs. Training is carried out for 3600 epochs. The margin parameter of the hinge function is set to $\tau = 5$, while the valid/invalid triples ratio is set to 2.

For our experiments we train the BERT LM [2] on the English Wikipedia corpus dataset[2], containing more than 6.4M text rows. Each rows contains the corpus of one full Wikipedia article, preprocessed to remove markdown and unwanted sections (such as hyperlinks and references). Additionally, we add node labels from WN18RR and FB15K-237 to the tokenizer's vocabulary.

Both the BERT-KG and BERT-BASELINE models were trained from scratch on Wikipedia corpus with the same configuration of parameters. We use 12 Transformer layers, each containing 12 attention heads. The model processed a batch of 8 sequences of length 512 at a time. The internal model size is 768. For BERT-KG, the regularisation weight is set to $\lambda = 10$. We decide to keep only the associated KG node embeddings that reach a similarity score threshold of 0.3. The models are trained for 800k steps using the AdamW optimizer with learning rate $lr = 5 \times 10^{-5}$.

[2] https://huggingface.co/datasets/wikipedia - 20200501.en

Table 2: Results of the regularised LM on the SQuAd knowledge completion probing task. MRR is the Mean Reciprocal Rank and P@10 is the Top-10 precision.

Model	MRR	P@10
`BERT-BASELINE`	0.070	0.172
`BERT-KG WN18RR`	0.071	0.174
`BERT-KG FB15k-237`	**0.074**	**0.184**

5 Probing Tasks

We run a series of experiments to compare `BERT-BASELINE` and the `BERT-KG` models trained on the WN18RR and FB15K-237 KGs. We use two *probing tasks* as a benchmark for the comparison.

The first task is based on the MEN Test collection dataset [16], containing 3000 pairs of words with an assigned similarity score. We want to predict the similarity score between the two word representations computed by BERT. In Table 1, we compare the result between `BERT-KG` and `BERT-BASELINE` models. We report Mean Squared Error (MSE), R^2 score and Spearman's correlation between the model's predictions and the ground truth. While the MSE is similar for all models, the R^2 score and the correlation scores are higher for the regularised models, especially for WN18RR. This suggests that the injection on general linguistic knowledge into the word embeddings is indeed beneficial for this task.

The second task is based on the Stanford Question Answering Dataset (SQuAD) [17], a collection of question-answer pairs derived from Wikipedia articles. Following the approach of [18], we use a subset of 305 context-insensitive questions from the SQuAD development set with single token answers. The LM is queried for factual knowledge using cloze tests.For this task we consider the Mean Reciprocal Rank (MRR) and the Top-10 precision (P@10). Results are reported in Table 2. In this case, the model regularised on FB15K-237 yields the highest performance. This is in line with our intuition, since SQuAD requires the type of knowledge that is contained in the FB15K-237 KG.

6 Conclusion

In this paper, we used KGs to enrich the representations learned by the BERT LM. We first matched the nodes of the KG with the corresponding words of the text corpus, and then we regularised the corresponding embeddings in order to make them similar to each other. The experimental results show that this approach is effective for injecting the information of the KG into the LM representations. In the future, we plan to explore multiple parallel KGs in the regularisation process. We also plan to include a more thorough evaluation of the approach by designing new types of probing tasks.

References

[1] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, et al. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509, 2015.

[5] Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 2020.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Wen Zhou, Haoshen Hong, Zihao Zhou, and SCPD Stanford. Derive word embeddings from knowledge graph.

[8] Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. *arXiv preprint arXiv:1906.01195*, 2019.

[9] Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124(3):1907–1922, 2020.

[10] Zhibin Lu, Pan Du, and Jian-Yun Nie. Vgcn-bert: augmenting bert with graph embedding for text classification. In *European Conference on Information Retrieval*, pages 369–382. Springer, 2020.

[11] Pedro L Rodriguez, Arthur Spirling, and Brandon M Stewart. Embedding regression: Models for context-specific description and inference. Technical report, Working Paper Vanderbilt University, 2021.

[12] Jingchao Ni, Shiyu Chang, Xiao Liu, Wei Cheng, Haifeng Chen, Dongkuan Xu, and Xiang Zhang. Co-regularized deep multi-network embedding. In *Proceedings of the 2018 World Wide Web Conference*, pages 469–478, 2018.

[13] Tingting Wei, Yonghe Lu, Huiyou Chang, Qiang Zhou, and Xianyu Bao. A semantic approach for text clustering using wordnet and lexical chains. *Expert Systems with Applications*, 42(4):2264–2275, 2015.

[14] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*, 1994.

[15] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[16] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47, 2014.

[17] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[18] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*, 2020.