Contextual Language Models for Knowledge Graph Completion

Russa Biswas, Radina Sofronova, Mehwish Alam, and Harald Sack

FIZ Karlsruhe - Leibniz Institute for Information Infrastructure,
Karlsruhe Institute of Technology, AIFB
firstname.lastname@fiz-karlsruhe.de

Abstract. Knowledge Graphs (KGs) have become the backbone of various machine learning based applications over the past decade. However, the KGs are often incomplete and inconsistent. Several representation learning based approaches have been introduced to complete the missing information in KGs. Besides, Neural Language Models (NLMs) have gained huge momentum in NLP applications. However, exploiting the contextual NLMs to tackle the Knowledge Graph Completion (KGC) task is still an open research problem. In this paper, a GPT-2 based KGC model is proposed and is evaluated on two benchmark datasets. The initial results obtained from the fine-tuning of the GPT-2 model for triple classification strengthens the importance of usage of NLMs for KGC. Also, the impact of contextual language models for KGC has been discussed.

Keywords: GPT-2 · Knowledge Graph Embedding · Triple Classification.

1 Introduction

Knowledge Graphs (KGs) such as DBpedia, YAGO, Freebase, etc. have emerged as the backbone of various applications in Natural Language Processing (NLP) such as entity linking [9], question answering [2], etc. KGs are multi-relational directed graphs with nodes as real world entities and relationships between them are represented on the edges. The facts are represented as a triple $< h, r, t >$, where h and t are the head and tail entities respectively and r represents the relation between them. However, these KGs are often incomplete. Knowledge Graph Completion (KGC) is the task of predicting the missing links between entities, mining missing relations, and discovering new facts. Recent years have witnessed extensive research on KGC with a focus on representation learning. Most of these models use structural information i.e., the triple information such as TransE [3], ConvE [5] whereas a few others include textual entity descriptions such as TEKE [22], DKRL [25], etc. However, the models considering the textual information leverage only static word embedding approaches, such as word2vec, GloVe etc. to generate the latent representation of the textual entity descriptions. Consequently, the semantic information encoded in the contextual entity embeddings are not exploited for KGC.

On the other hand, pre-trained contextualized Neural Language Models (NLMs) such as BERT [11], GPT-2 [20], have gained huge momentum in applications of NLP. These models are trained on huge amount of free text resulting in encoding of the semantic information leading to better linguistic representation of the words. GPT-2 is one of the distinguished models which has achieved state-of-the-art results for various language understanding based tasks. It operates on a transformer decoder architecture with attention masks to predict next word of a sequence.

However, a combination of contextualized NLMs for the task of KGC is an open research problem. KG-BERT [29] is one of the pioneers in this research in which the BERT model is fine-tuned on KG data and has been used for link prediction and triple classification as sub-tasks of KGC. The results presented in [29] depict that the information contained in pre-trained NLMs play an important role in the predicting the missing links in a KG. Inspired by KG-BERT, a novel GPT-2 based KGC model is explored in this work for the triple classification sub-task. The triples in a KG are considered as sentences and the triple classification is considered as a sequence classification problem. Furthermore, an analysis of the contextualised NLMs for KGC is also provided.

The rest of the paper is organised as follows. To begin with, a review of the related work is provided in Section 2 followed by the preliminaries in Section 3. Section 4 accommodates the outline of the proposed approach followed by experimental results in Section 5. Finally, an outlook of future work is provided in Section 6.

2 Related Work

This section presents the state-of-the-art (SOTA) models for KG embeddings with a focus on the models considering the textual descriptions.

A large variety of KG embedding approaches has been explored for the task of link prediction, such as translational models like TransE [3] and its variants, semantic matching models like DistMult [28], neural network based models like ConvE [5], graph structure based like GAKE [6], and literal (e.g., text, image, number, etc.) based like DKRL [25], Jointly(ALSTM) [27], MKBE [17], etc.

In a translational model such as TransE [3], given a triple (e_h, r, e_t) in a KG G, the relation r is considered as a translation operation between the head and tail entities on a low dimensional vector space defined by $\mathbf{e_h} + \mathbf{r} \approx \mathbf{e_t}$, where $\mathbf{e_h}, \mathbf{r}, \mathbf{e_t}$ are the embeddings of the head, relation and the tail entity respectively.

Another set of algorithms improve KG embeddings by taking into account different kinds of literals such as numeric, text or image literals and a detailed analysis of the methods is provided in [7]. DKRL [25] extends TransE [3] by incorporating the textual entity descriptions in the model. The textual entity descriptions are encoded using a continuous bag-of-words approach as well as a deep convolutional neural network based approach. Jointly(ALSTM) is another entity description based embedding model which extends the DKRL model with a gate strategy and uses attentive LSTM to encode the textual entity descrip-

tions. KG-BERT [29] is a contextual NLM based model which is fine tuned on BERT and have been used in downstream tasks.

However, the contextual NLMs are not considered to encode the triples or the entity descriptions in all the models except KG-BERT. Therefore, this study proposes a novel model in which the KG is fine-tuned with GPT-2 for KGC.

3 Preliminaries

A detailed explanation of pre-trained NLMs and KGC is provided in this section.

3.1 Language Models

A LM learns the probability of word occurrences based on a text corpus which is used for various machine learning based NLP applications such as Machine Translation [12], Speech Recognition [30], etc. It is the task of assigning probability to each sequence of words or a probability for the likelihood of a given word based on a sequence of words. [8]. LMs can be broadly divided into

- **Statistical Language Models (SLMs)** are *n-gram* based approaches that assign probabilities to a sequence s of n words, and is given by

$$P(s) = P(w_1 w_2 ... w_n) = P(w_1)P(w_2|w_1)...P(w_n|w_1 w_2 ... w_{(n-1)}), \quad (1)$$

  where w_i denotes $i-$th word in the sequence s. The probability of a word sequence is the product of the conditional probability of the next word given the previous words or the context [10]. The SLMs fail to assign probabilities to the n-grams that do not appear in the training corpus which is tackled using the smoothing techniques. However, the curse of dimensionality refrains the SLMs models to be trained on huge corpora.
- **Neural Language Models (NLMs)**, on the other hand, are neural network based LMs that learn the distributed representation of words into a continuous low-dimensional vector space. The semantically similar words appear closer to each other in the embedding space. The contextual information is captured on all the different levels in the text corpus, such as, sentences, sub-word, character, as well as the entire corpus.

The NLMs such as Word2Vec [13], BERT [11], GPT [19] etc. are beneficial for several NLP downstream tasks, such as question answering [23], sentiment analysis [26], etc. As mentioned in [18], these models can be further sub-divided into **(i)** Non-contextual and **(ii)** Contextual Embeddings. The Non-contextual word embeddings such as Word2Vec, GloVe, etc., are static in nature and are context independent. Although, the latent representations of the words capture the semantic meanings but they do not dynamically change according to the context the words appear in. However, Contextual embeddings such as BERT, GPT, etc., encode semantics of the words differently based on different contexts. All the language models are trained on huge unlabelled text corpora resulting in

increased number of model parameters. Therefore, the pre-trained models help in learning universal language representations of the words. It promotes better initialization of the model to have a better generalization performance on the downstream tasks. Pre-training of the NLMs also helps in avoiding overfitting of the model for small corpora [18]. Also, it improves the reuseability of the model as it prevents the training of the model from scratch. However, fine tuning of pre-trained contextual NLMs is often required to adapt the model to the specific data for the down-stream task. It bridges the gap between the data on which a particular NLM is trained on and the target data distribution.

3.2 Knowledge Graph Completion

The goal of KGC is the task of predicting missing instances or links to deal with the incompleteness and sparsity in KGs. As explained in [4] KGC methods can be broadly divided into the following classes:

- **Rule Based Models** that use rules or statistical features such as NELL [15], KGRL [24], etc., to infer new knowledge in KGs.
- **Representation Learning Based Models** such as TransE [3], ConvE [5], etc., that learn the latent representation of the entities and relations into a low-dimensional continuous vector space, in which semantically similar entities are placed closer to each other. These representations are then used for the KGC tasks of link prediction and triple classification.

In link prediction task, the head or tail entity in a triple $< h, r, ? >$ or $<?, r, t >$ is predicted by defining a mapping function $\psi : E \times R \times E \rightarrow R$, where E and R are the set of entities and relations in the KG. A score is assigned to each triple, where the higher the score of the triple indicates the more likely to be true. The triple classification task involves the training of binary classifier whether a given triple is false (0) or true (1).

4 Language Models for Knowledge Graph Completion

This section comprises of an analysis of NLMs on KGs followed by a detailed description of the GPT-2 based KGC task. The basic idea of the approach lies in the fact that the contextual NLMs trained on huge corpora also capture relational information present in the training data [16]. Consequently, NLM models can be exploited further to predict the missing links in a KG. However, the impact of the pre-trained contextual NLMs for KGC is still an open research.

BERT for KGC One of the pioneers in this domain is the KG-BERT [29] model in which the pre-trained BERT model is fine-tuned on KGs for KGC. Each triple $< h, r, t >$ is considered as a sentence and is provided as an input sentence of the BERT model for fine-tuning. For the entities, KG-BERT has been trained with either the entity names or their textual entity descriptions.

The first token of every input sequence is always $[CLS]$, whereas the separator token $[SEP]$ separates the head entity, relation and the tail entity. Therefore, each input sequence for the BERT model is given by
([CLS] head entity/description [SEP] relation [SEP] tail entity/description [SEP]).
A sigmoid scoring function is introduced on the top of the final layer for the triple classification which is a 2-dimensional vector $\in [0, 1]$.

GPT-2 for KGC Inspired by KG-BERT, GPT-2 [20] is exploited in this work for KGC. GPT-2 is a large transformer-based language model trained on 8 million web pages with 1.5 billion parameters. The model predicts the next word based on all the previous words in the text corpus. An attention mechanism is used to selectively focus on the segments of the input text. The architecture comprises of a 12-layer decoder-only transformer, using 12 masked self-attention heads, with 64 dimensional states each. The Adam optimization is used and the learning rate was increased linearly from zero to a maximum of 2.5×10^{-4}. The model was able to outperform the previous NLMs on language tasks like question answering, reading comprehension, summarization, translation, etc. However, the basic difference between BERT and GPT-2 is that BERT uses transformer encoder blocks whereas GPT-2 uses transformer decoder blocks.

Similar to KG-BERT, GPT-2 is also fine tuned with KG triples where each triple is considered as an input sequence. In this model, two variants have been used to model the input sequence for the fine-tuning task. Given a triple *Albert Einstein, bornIn, Germany*, the input sequence is modelled as

- Albert Einstein bornIn Germany $[EOS]$,
- $[BOS]$ Albert Einstein $[EOS]$ bornIn $[EOS]$ Germany $[EOS]$,

where *[BOS]* and *[EOS]* are the beginning of sequence and end of sequence respectively. Both entity names and descriptions are considered for the head and tail entity. The input sequences are fed into the GPT-2 model architecture which is a transformer decoder based on the original implementation [20]. It consists of stacked decoder blocks of the transformer architecture and the context vector is initialised with zero for the first word embedding. The masked self-attention is used to extract information from the prior words in the sentence as well as the context word. The word vectors in the first layer of GPT-2 follows byte pair encoding i.e., tokens are parts of words. Furthermore, it compresses the tokenized words list into a set of vocabulary items by considering the most common word components. The GPT-2 sequence classification module is leveraged to determine the plausibility of the triples. Since, GPT-2 outputs one token at a time, the classifier is built on the last token. A 2-dimensional vector $\in [0, 1]$ sigmoid scoring function is introduced for triple classification.

5 Experiments

This section comprises of an analysis of the initial results obtained on deploying GPT-2 model on the triple classification task for KGC. The model has been evaluated on two benchmark datasets WN11 and FB13.

Table 1. Dataset Statistics

Dataset	#Ent.	#Rel.	#Train	#Val.	#Test
WN11	38,696	11	112,581	2,609	10,544
FB13	75,043	13	316,232	5,908	23,733

Table 2. Results of Language Models on Triple Classification (accuracy in %)

Model Types	Models	WN11	FB13
KG embeddings with Textual	TEKE	86.1	84.2
Contextual LMs	KG-BERT (labels)	93.5	79.2
	KG-BERT (description)	-	90.4
	Ours with GPT2 (labels)	83	73
	Ours with GPT2 (description)	85	89

Datasets The two benchmark datasets WN11 and FB13 are subsets of WordNet and Freebase KGs respectively and are introduced in [21]. WordNet [14] is a large lexical KG of English comprising of nouns, verbs, adjectives and adverbs. They are grouped into sets of cognitive synonyms known as synsets. Each synset expresses a distinct concept. They are interlinked by means of conceptual-semantic and lexical relations. Freebase [1] is a large collaborative KG consisting of structured data captured from various sources including individual, user-submitted wiki contributions. The statistics of the KGs used to fine-tuning with GPT-2 followed by triple classification is provided in Table 1.

Experimental Setup The pre-trained GPT-2 base model with 12 decoder layers, 768 hidden layers, 12 attention heads and 117M parameters is used for fine-tuning. The set of hyperparameters chosen are as follows: batch sizes = $\{256, 128, 32, 8, 1\}$, epochs = $\{5, 3\}$, and learning rate = $\{2e-5, 5e-5\}$. The experiments with GPT-2 have been performed on an Ubuntu 16.04.5 LTS system with 503GB RAM and Tesla V100S GPU.

Results The results depicted in Table 2 represent some initial results on the triple classification task using the pre-trained GPT-2 model on KGs. Since all the triples in the training set are true, a negative sampling method is used to generate synthetic negative triples for the training of the classifier. The negative triples are generated for this task, by replacing the head and the tail entities with arbitrary entities based on a local closed world assumption. In this work, filtered settings is used, i.e., if by chance true triples are generated using negative sampling methods, then they are removed. Therefore, the set of triples in the train, test, and validation sets are disjoint.

TEKE [22] and KG-BERT are considered as baseline models as they consider NLMs to model the KGs for KGC. TEKE exploits structural information of the KGs using an embedding layer, a BiLSTM layer followed by mutual attention

Table 3. Results with the pre-trained GPT2 model for Triple Classification with different parameter settings

Dataset	Feature	Model details	Precision	Recall	F_1-score
WN11	Labels	batch=128, epoch=10, lr=2e-5	0.76	0.76	0.76
		batch=32, epoch=3, lr=5e-5	0.74	0.74	0.74
		batch=1, epoch=3, lr=5e -5	0.83	0.83	0.83
	Description	batch=8, epoch=5, lr=2e - 5	0.79	0.79	0.79
		batch=1, epoch=3, lr=5e -5	0.85	0.85	0.85
FB13	Labels	batch=32, epoch=10, lr=2e -5	0.69	0.64	0.61
		batch=256, epoch=5, lr=2e -5	0.68	0.68	0.68
	Description	batch=1, epoch=3, lr=5e-5	0.90	0.89	0.89

layer. The results of the baselines are taken from the KG-BERT paper [29] except for KG-BERT (labels) variant for FB13. The experiment for this variant is performed with the same settings as mentioned in [29]. It is observed from the results that with GPT-2, the model achieves comparable results with the previous models. Also, the results are better for GPT-2 with descriptions variant, this is because the textual entity descriptions have more contextual information resulting in generation of better representation of triples. The same behaviour has been observed for KG-BERT. Since the NLMs are trained on large corpora, the model parameters contain huge amount of linguistic knowledge which helps in overcoming the data sparsity problem in KGs. Furthermore, the main advantage of contextual NLM based KGC methods that they do not consider the structural information of the entities in a KG. Hence it is independent of any underlying structure in a KG. Furthermore, these models are also applicable to the less popular entities in KGs with lesser number of triples compared to the others. The task of triple classification in KGC with GPT-2 is similar to the sequence classification task in text and the self attention mask helps in identifying the important words in the sequences. The variants with labels i.e., the entity names for both KG-BERT and the proposed GPT-2 based model work better for WN11 as compared to FB13. This is because WordNet is a linguistic KG and the NLMs are able to capture more information on the entity names as compared to FB13.

Table 3 depicts the precision, recall, and F_1 score of the model with different hyper-parameter settings. It is observed that the best results are obtained with batch=1, epoch=3, and lr=5e-5. The changing of epochs does not have much variation in the model whereas batch size has. The lower the batch size, the better the performance of the model.

6 Conclusion and Future Work

This work presents an analysis of the effect of exploiting NLMs for KGC. A novel GPT-2 based KGC model has also been proposed. The initial results from the triple classification sub-task shows that the semantic information stored in the NLMs can provide vital information for the KGC task. In future, further hyper-

parameter tuning will improve model performance and additional experiments on link prediction sub-tasks will be conducted.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250 (2008)
2. Bordes, A., Chopra, S., Weston, J.: Question answering with subgraph embeddings. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 615–620 (2014)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems **26** (2013)
4. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge graph completion: A review. IEEE Access **8**, 192435–192456 (2020)
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Thirty-second AAAI conference on artificial intelligence (2018)
6. Feng, J., Huang, M., Yang, Y., Zhu, X.: GAKE: Graph aware knowledge embedding. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 641–651. The COLING 2016 Organizing Committee, Osaka, Japan (Dec 2016), `https://www.aclweb.org/anthology/C16-1062`
7. Gesese, G.A., Biswas, R., Alam, M., Sack, H.: A survey on knowledge graph embeddings with literals: Which model links better literal-ly? Semantic Web (Preprint), 1–31
8. Goldberg, Y.: Neural network methods for natural language processing. Synthesis lectures on human language technologies **10**(1), 1–309 (2017)
9. Hoffart, J., Yosef, M.A., et al., I.B.: Robust disambiguation of named entities in text. In: Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2011. pp. 782–792 (2011)
10. Jing, K., Xu, J.: A survey on neural network language models. arXiv preprint arXiv:1906.03591 (2019)
11. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. pp. 4171–4186 (2019)
12. Koehn, P.: Statistical machine translation. Cambridge University Press (2009)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
14. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
15. Paulheim, H., Bizer, C.: Improving the quality of linked data using statistical distributions. International Journal on Semantic Web and Information Systems (IJSWIS) **10**(2), 63–86 (2014)
16. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: 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). pp. 2463–2473 (2019)
17. Pezeshkpour, P., Chen, L., Singh, S.: Embedding multimodal relational data for knowledge base completion. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 3208–3218 (2018)
18. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: A survey. Science China Technological Sciences pp. 1–26 (2020)
19. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
21. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. pp. 926–934 (2013)
22. Wang, Z., Li, J., Liu, Z., Tang, J.: Text-enhanced representation learning for knowledge graph. In: Proceedings of International Joint Conference on Artificial Intelligent (IJCAI). pp. 4–17 (2016)
23. Wang, Z., Ng, P., Ma, X., Nallapati, R., Xiang, B.: Multi-passage bert: A globally normalized bert model for open-domain question answering. 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). pp. 5878–5882 (2019)
24. Wei, Y., Luo, J., Xie, H.: Kgrl: an owl2 rl reasoning system for large scale knowledge graph. In: 2016 12th International Conference on Semantics, Knowledge and Grids (SKG). pp. 83–89. IEEE (2016)
25. Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30 (2016)
26. Xu, H., Liu, B., Shu, L., Philip, S.Y.: Bert post-training for review reading comprehension and aspect-based sentiment analysis. 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). pp. 2324–2335 (2019)
27. Xu, J., Qiu, X., Chen, K., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1318–1324 (2017)
28. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), `http://arxiv.org/abs/1412.6575`
29. Yao, L., Mao, C., Luo, Y.: Kg-bert: Bert for knowledge graph completion. arXiv preprint arXiv:1909.03193 (2019)
30. Yu, D., Deng, L.: Automatic Speech Recognition. Springer (2016)