Knowledge Graph Completion using Structural and Textual Embeddings*

Sakher Khalil Alqaaidi[1] and Krzysztof Kochut[1]

School of Computing, University of Georgia
{sakher.a,kkochut}@uga.edu

Abstract. Knowledge Graphs (KGs) are widely employed in artificial intelligence applications, such as question-answering and recommendation systems. However, KGs are frequently found to be incomplete. While much of the existing literature focuses on predicting missing nodes for given incomplete KG triples, there remains an opportunity to complete KGs by exploring relations between existing nodes, a task known as relation prediction. In this study, we propose a relations prediction model that harnesses both textual and structural information within KGs. Our approach integrates walks-based embeddings with language model embeddings to effectively represent nodes. We demonstrate that our model achieves competitive results in the relation prediction task when evaluated on a widely used dataset.

Keywords: Knowledge Graphs · Text Mining.

1 Introduction

Knowledge Graphs (KGs) store informational facts in a structured format of connected triples. A triple consists of a semantic relation that binds a head (subject) entity node with a tail (object) entity node. KGs have been widely employed in several artificial intelligence applications on a global data scale, such as question-answering [33] and recommendation systems [24]. Due to their enormous sizes [13,2], Knowledge Graphs suffer from being incomplete [32]. Given that the KG incompleteness problem affects its utilization, several works proposed Knowledge Graph Completion (KGC) models. These works tackled three types of KGC, that are Link Prediction (LP), Relation Prediction (RP), and Triple Classification (TC). The LP task aims at finding the missing head or tail node in a triple. The RP task aims at finding the valid relations for a given pair of head and tail nodes. The TC task aims at determining the plausibility of a complete triple.

After surveying the literature, we found that much of the literature works focused on the LP task, whereas the RP task remained a secondary objective in some models [23]. A few models proposed an RP variant based on a link

* Accepted in AIAI 2024 - 20th International Conference on Artificial Intelligence Applications and Innovations

prediction-oriented design [14]. However, we see that the RP task holds equal importance to the LP task and can effectively contribute to KGC. That is by identifying all the possible relations between node pairs. For instance, if *Apple* and *California* are two nodes in a knowledge graph and connected using the relation *has_store_in*, we could be interested in exploring other possible relations between them. For example, one relation that can be established between them is *headquarters_in* after identifying *Apple* as the node for the technological company.

Recently, KGC models have achieved remarkable accomplishments by representing graphs in the form of multi-dimensional float vectors [23]. Therefore, KGC models can be categorized into three groups according to the representation type used. First, KGC models utilized the graph's structural information [3,25]. Second, KGC models utilized the nodes' meta information [34]. Third, KGC models utilized a combination of the two information types [32,29,35]. In terms of meta information utilization, several models have used the textual content within node entity names as input for language models, enabling the generation of representation embeddings. This approach has demonstrated efficacy, particularly in inductive settings [6], that is when prediction is conducted for nodes not seen during model training. Notably, Masked Language Models (MLMs) showed astonishing results in the KGC task using nodes' text. However, these language models suffer from a costly fine-tuning phase in terms of computational resource usage, such as time and memory. In contrast, Pre-trained Language Models (PLMs) have significantly lower computational load and still provide meaningful representation of words without training the model again.

Although language models cap present powerful text representation, employing the graph structural information is still crucial in the KGC task [29], primarily due to the entity ambiguity problem encountered when encoding text content. Particularly, because language models cannot represent an entity efficiently relying on short text. Similar to the previous *Apple* and *California* example, language model embeddings for the *Apple* node cannot lead to determining whether it is the technological company or just the fruit without having additional clues other than the node name. Otherwise, the relation *planted_in* would be predicted because *Apple* could also be a node for the fruit. Indeed, determining the node identity would be possible only if the text content is detailed as in this sentence: *Apple is a technology company headquartered in California.* However, such detailed description is not usually provided in KGs without employing connections to external knowledge resources. In contrast, the node's structural details can support revealing the node's identity depending on topological information, such as the neighborhood and the walks obtained from there [19].

To this end, we propose a **r**elations **p**rediction model (RPEST) that utilizes the **e**mbeddings of **s**tructural and **t**extual features in knowledge graphs. Our model employs a walk-based graph structure algorithm, namely Node2Vec [9], to replace the language model fine-tuning step with structural embeddings training. Additionally, we exploit pre-trained language models efficiently to capture text contextualized representation and to avoid the costly fine-tuning phase

in MLMs. We show that our choice of language model has significantly lower overhead compared to masked language models. Notably, our model achieves competitive results in the relation prediction task when compared to the best accomplishments in Freebase, a widely used benchmark. Furthermore, we analyze the effectiveness of our model's components in a dense ablation study. To the best of our knowledge, this is the first work that considers textual embeddings along with graph walks-based embeddings for the RP task. The code to reproduce the results is available online [1]

2 Background and Related Work

Given a knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{E}$ is a set of entity nodes, $\mathcal{R}$ is a set of relations of size k, and $\mathcal{T}$ is a set of entities text. $\mathcal{G}$ is constructed of facts; each fact is represented as a directed triple (h, r, t), where $h, t \in \mathcal{E}$, h is a head entity node, t is a tail entity node, $r \in \mathcal{R}$, and r is a relation with a direction from h to t; the direction is essential to hold the fact, thus (t, r, h) is not equivalent to (h, r, t). The relation prediction task targets finding the probability of all relations in $\mathcal{R}$ for the given nodes h and t, formally:

$$\phi(h, t) = Pr(\mathcal{R}) \tag{1}$$

where ϕ is a probability values scoring function. In contrast, the link prediction task aims to find a missing head node $(?, r, t)$ or a missing tail node $(h, r, ?)$ in a triple. Recent achievements in knowledge graph completion have followed a representation learning approach [4]. Specifically, models targeted learning a function that can efficiently represent knowledge graph triples in a multi-dimensional float values space $\mathbb{R}$, i.e., continuous vector embeddings. KGC models can be categorized into methods based on the node's structural information, textual content information, or a combination of both.

2.1 Graph Structural Approaches

Several translation-based methods have achieved remarkable results in the link prediction task [25,3,15]. TransE [3] started a translation-based stream of methods for unsupervised graph embeddings learning. The translation technique uses distance-based scoring function to find close embedding values of head and tail nodes given the relation between them. Specifically, finding the minimum value of the embeddings of the three triple elements in this equation $|h + r - t|$, where h, r, and t are represented in a d-dimensional space $\mathbb{R}^d$, i.e., $h, r, t \in \mathbb{R}^d$. Since TransE used a unified space for nodes and relations. TransR [15] proposed a multi-relational spaces model to tackle the problem of nodes involved in multiple facts in the knowledge graph. Formally, each relation is represented in the space $\mathbb{R}^k$, and the knowledge graph facts are represented in the space $\mathbb{R}^{k \times d}$.

[1] https://github.com/sa5r/KGRP

However, this approach came at the cost of additional computation due to the extended spaces. TaRP is a relation prediction model [5]. TaRP employed different translation-based methods such as TransE and RotateE, to generate representation embeddings. Additionally, TaRP incorporated the node type identification and encoding to boost the relation prediction results. A different course to generate node embeddings is using neural networks. Shallom [7] proposed a shallow neural network layers for the relation prediction task.

2.2 Textual Content Approaches

The textual content in KGs has been exploited in different ways. The node's long description helped in predicting the relations for nodes that were not seen during model training, i.e., inductive settings [26]. However, retrieving each node's text description requires additional steps and the employment of external knowledge bases.

BERT [8] emerged as a masked language model with a state of the art performance on a variety of Natural Language Processing (NLP) tasks KG-BERT [34] was one of the first models to utilize BERT for the KGC task; KG-BERT targeted mainly the LP task. Additionally, two variants were proposed for the RP and TC tasks. Inspired by this work, BERTRL [36] linearized the triples around each relation and trained the model to have better inductive performance. Nevertheless, language model-based approaches suffered from serious issues. BERT and its successors of masked language models [16], incorporate a costly fine-tuning phase in terms of time and memory. In large language models (LLMs) such as GPT and Llama, the computation needs are worst [27,20].

In contrast, pre-trained language models (PLMs) have significant lower computational load. Furthermore, PLMs provide meaningful representation of text words for new datasets without retraining the model. For instance, Glove [18] is a PLM that provides multi-dimensional vectors representation for a single word with $O(n)$ complexity, where n is the PLM's vocabulary size. However, Glove operates at the word level, unlike MLMs and LLMs that can represent text sequences as contextualized embeddings. Text sequences can consist of mutli-word terms or sentences. A node's text content can include multiple words, such as 'Alfred Nobel'.

2.3 Hybrid Approaches

Here we review KGC models that employed a hybrid approach based on textual and structural representation. LASS model [22] employed BERT's fine-tuning for the textual encoding and used the resulted loss values to construct the semantic embeddings. A variant for triple classification was proposed in the paper. StAR model [29] targeted handling the overwhelming performance in language models by reusing the graph elements' embeddings. However, the model achieved a satisfactory performance only after combining the its textual representation with an existing graph embeddings, such as RotateE. The KGLM model [35]

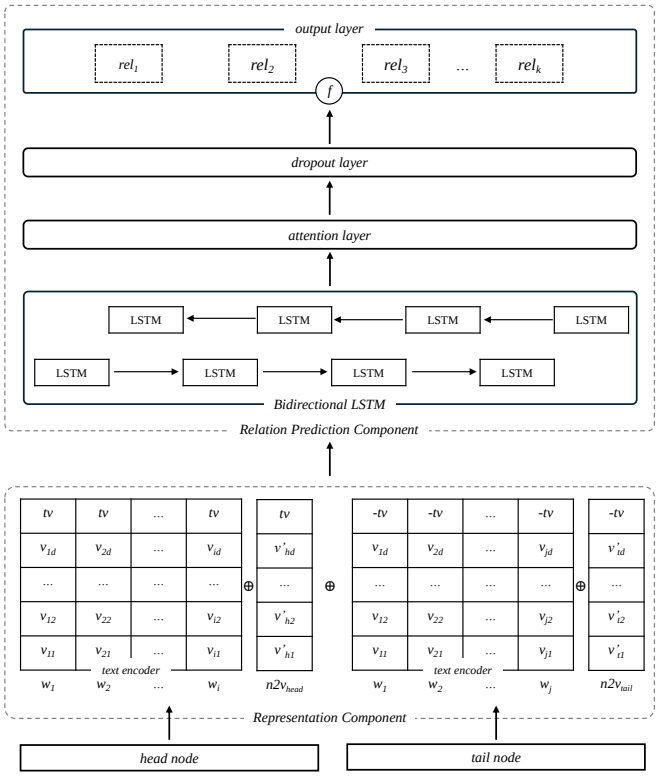

Fig. 1. Our model's diagram showing the node representation component and the relations prediction component.

re-ran the original Roberta language model training with additional text generated from translating KG triples into normal sentences. The model added an entity/relation-type embedding layer to include the graph structure in training. In KEPLER [31], authors jointly optimized the language model training objective with the structural training objective for the link prediction task.

Some models achieved better results in the link prediction task through trying different negative sampling (contrastive learning) approaches in the text embeddings training [30,11]. The SimKGC model [30] incorporated the structured information by re-ranking candidates based on path distances between target nodes in the graph, whereas the MoCoSA model [11] used the independently trained structural embeddings and fused them with the textual embeddings.

3 Methodology

Our model aims to find the probability of every predefined relation that can establish a fact by connecting a head node to a tail node. Formally, $Pr(\mathcal{R}|h,t)$

where $\mathcal{R}$ is a set of predefined relations, h is the head node, and t is the tail node. Our model takes as an input a combination of the node structural representation and the node textual representation. Our model conducts a supervised learning for a neural network to predict the relations' probabilities. Accordingly, our model consists of two main components, the representation component and the relations prediction component. The former incorporates our approach to generate the structural representation and the textual representation, whereas the later incorporates a recurrent neural network and the prediction layer. Figure 1 shows the architecture of our model.

3.1 Structural Representation Training

We incorporate the nodes structural information using the Node2Vec [9] model. Node2Vec has been known for its efficiency and for being versatile. The model is an unsupervised learning algorithm for graph features representation. Training in Node2Vec aims at optimizing a function that maximizes the log-probability of observing a neighborhood for a node, based on the neighborhood nodes' representation as the following:

$$\max_{f} \sum_{u \in V} log\, Pr(N_s(u)|f(u))$$

where V is the nodes set in a graph $\mathcal{G}$, $N_s(u)$ is the neighborhood for a node u, and $f(u)$ is the features representation for the node u. The source node neighborhood is constructed from sampling nodes identified following biased random walk strategies. The strategies could be either Breadth-First Sampling (BFS) or Depth-First Sampling (DFS). In the former, the sampled nodes are adjacent to the source node, whereas in the later, the sampled nodes are found while taking further steps in the walk paths, so it is not necessary for the nodes to be immediately connected to the source node, but a direct path should lead to the source node.

Node2Vec is used in our model as a preliminary step before training the prediction neural network, that is described in section 3.3, this initial phase replaces the fine-tuning step in MLMs, which is not needed in our implementation as described in section 3.2. After training the Node2Vec algorithm on the graph, we get a representation $v^{'}$ of d vectors for each node as shown in Figure 1.

3.2 Text Encoding

We employ Glove [18] pre-trained word embeddings to represent the nodes' textual content. Glove language model is an unsupervised learning algorithm trained on a word-word co-occurrence probabilities using a global scale text corpus. The training objective in Glove is minimizing the difference between two words co-occurrence as the following:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w_j} + b_i + \tilde{b_j} - log\, X_{ij})^2$$

where V is the number of words in Glove's vocabulary, i and j are identification numbers for two words, w_i and b_i are the vector and the bias for word i, w_j and b_j are the vector and bias for word j, X_{ij} is the number of times word i occurred in the context of word j, and f is a weighting function for the co-occurrences.

We chose Glove due to it is superiority among other language models, such as the skip-gram and continuous bag-of-words (CBOW) [17], and for providing immediate text representation without fine-tuning. However, Glove works at the word level, accordingly, the pre-trained embeddings E^d represent each word in Glove's vocabulary with vectors of size d dimensions. In our implementation, Glove encodes each node as $n \times d$ two-dimensional vector matrix, where n is the number of words to be considered in each node. When the number of words in a node is greater than n, we omit the words of index $(n + 1)$ and greater. In contrast, for nodes with number of words below n, we append embeddings of zeros to keep the embeddings consistent; each padding embedding has d vectors.

KG node names usually consist of multiple words. For instance, in Freebase [2], the average entity name length is 2.7 words. Although multi-word terms and sentences can be represented efficiently in MLMs and LLMs, PLMs still can present a contextualized encoding by employing a Recurrent Neural Network (RNN). Consequently, we employ a bidirectional long-short term memory (LSTM) layer in our model's neural network. RNNs are capable of encoding the latent features in text sequences [21]. For the Out-of-Vocabulary (OOV) words problem [37], we find the longest possible match for a word in Glove's vocabulary or take the word letter embeddings average in the worst case. However, OOV words still occupy less than 2% for most datasets when using Glove, due to its large vocabulary.

We enhance the recognition of the relation direction in the input by appending additional vector at the top of each node's representation, this approach has been proven to help predicting the proper relations based on the direction [1]. The head node uses a fixed value in the added vector and the tail node uses the same value but multiplied by -1.

3.3 Relation Prediction

The architecture of our model's neural network ensures the utilization of the latent features in the combined node embeddings. The input embeddings for each pair of nodes consist of a head node representation concatenated with a tail node representation. Each node's representation consists of the textual embeddings concatenated with the structural embeddings as shown in Figure 1. For each pair of nodes, the bidirectional LSTM layer generates new contextualized embeddings, that are forwarded to an attention layer to selectively focus on certain parts of the input data, allowing the model to weigh different inputs differently based on their relevance to the task [28]. In the following dropout layer [12] we enhance the generalization performance. Ultimately, the output layer uses the Sigmoid function [10] to give the probability of each relation. Our

Table 1. The training settings.

Settings	Value
Training Epochs	50
Early stopping epochs	5
Batch size	32
LSTM features in the hidden state	400
LSTM recurrent layers	2
Dropout probability	15%
Learning rate	0.0008
Optimizer decay	35%
Node representation padding	40
Node2Vec walk length	50
Node2Vec node walks	50

model computes the neural network training loss using the cross entropy loss function.

4 Experiments

We evaluated our model on FB15K, a subset of the FreeBase dataset [2]. FreeBase has been maintained by Google for several years and has been widely used for KG-related tasks; its content includes data collected from several sources such as Wikipedia. The last release of the complete dataset holds about 1.9 billion triples [2]. The FB15K subset holds 1,345 relations and 14,951 entities. The number of triples in the training portion is 483,142, the validation portion has 50,000 triples, and the testing portion has 59,071 triples.

We used PyTorch to train our model. The used hardware consisted of NVIDIA A100-SXM-80GB GPU node, a main memory of 50GB, and AMD EPYC MILAN (3rd gen) CPU processor; we utilized 8 cores of the CPU. Table 1 shows our training settings. We used Glove 300 dimensional word embeddings trained on 6 Billion tokens with a 400,000 words vocabulary. For the Node2Vec training, we used balanced settings in the biased walks. Specifically, we used equal parameters for the breadth-first and the depth-first sampling. For the model training optimization, we employed Adam algorithm. In the following sections we explain the evaluation metrics, the comparison models, the main results analysis, and the model components ablation study.

4.1 Evaluation Metrics

We evaluate our model on two commonly used metrics. First, the mean rank of the ground truth relation in the predicted relation probabilities set. Items in the set follow a descending order. Accordingly, the lower the mean rank the better the result. Since any pair of nodes may incur multiple correct relations, the rank

[2] *https://developers.google.com/freebase/*

Table 2. The results of our model and the comparison models on the FB15K dataset.

Model	Mean Rank	Hits@1	Filtered Mean Rank	Filtered Hits@1
TransE [3]	-	65	2.5	84
TransR [15]	-	42	2.1	91
KG-BERT [34]	1.69	69	1.25	**96**
Shallom [7]	1.59	73	1.26	95
Ours	**1.53**	**74**	**1.18**	94

of a correct relation should be decremented by the number of the valid relations that appeared on top of the correct one. This evaluation behaviour is called the filtered settings [3] and is reported in our experiments along with the raw mean rank. The second evaluation metric is called Hits@1, that is the ratio of ground truth relations appeared as the first item in the ordered relations probability values. We also report the filtered settings for this metric.

4.2 Comparison Models

We perform a comparison with models that used the structural information and the textual information in the RP task. Although some models showed significant results in the same task, we excluded these results from our comparison because of the utilization of node information other than the node name, such as the description or the node entity type. For instance TaRP [5] employed the node type information to achieve the reported performance. Nevertheless, most datasets do not have the node type information provided by default and that requires external knowledge bases with added load to retrieve and encode the type information. Additionally, TaRP did not generate embeddings for nodes but instead depended on existing models to do so. Similarly, the work in [26] used the entity description of long text to achieve similar results. In the following we show the models we consider in our comparison.

TransE [3] was originally evaluated on the link prediction task. However, several works used the method for the RP task, and we use the reported rank and hits results in our comparison. TransE utilized the structural information to represent nodes. TransR [15] was evaluated on the link prediction and triple classification tasks. Similarly, other works reported TransR's performance in the RP task. TransR also utilized the structural information to represent nodes. KG-BERT [34] reported their relation prediction performance in the FB15K dataset. As described in Section 2, the model's main design targeted the link prediction task. KG-BERT was also evaluated on the triple classification task. Shallom [7] tackled the RP task and utilized neural networks embedding layers to represent nodes.

4.3 Main Results

We show our main results in Table 2. The results show superiority of our relation prediction model compared to the best results in other RP models. However, KG-

Table 3. The results of RPEST variants on the FB15K dataset.

Model	Mean Rank	Hits@1	Filtered Mean Rank	Filtered Hits@1	Epoch Time$_{min}$
RPEST$_{\text{Glove}}$	1.63	68	1.26	**94**	**5**
RPEST$_{\text{BERT}}$	1.70	67	1.31	92	13
RPEST	**1.53**	**74**	**1.18**	**94**	6

BERT has equivalent filtered Hits@1 score. Nevertheless, our model beats the scores of the same model on the remaining metrics with good margin. The mean rank scores for TransE and TransR were not reported or found in any other study. We reason our model's superiority by the combination of the structural and textual details for every node. More precisely, we find the efficient utilization of the textual content leading the performance to this level as we show in the ablation study.

4.4 Ablation Study

To evaluate the effectiveness of our model's units, we created two variants to observe the model results after excluding some of its modules. Particularly, we created a variant that does not employ the nodes' structural information in the input. Therefore, we excluded the usage of Node2Vec in our model and we only kept Glove text encoder by creating a variant named RPEST$_{\text{Glove}}$. Furthermore, we evaluated our choice of language models by replacing Glove with BERT language model in a variant named RPEST$_{\text{BERT}}$. We show the evaluation metric scores of the different variants in Table 3. The table also shows the epoch time in minutes for each variant, and we rely on the time values to emphasize the performance difference.

The lower scores of the RPEST$_{\text{Glove}}$ variant proofed the importance of employing the structural information in representing nodes. However, the variant had the fastest performance due to the reduced embeddings size after excluding Node2Vec vectors. On the other hand, the Glove variant scored better results compared to the RPEST$_{\text{BERT}}$ variant. We reason that by the efficient exploitation of the latent features in Glove's embeddings using the bi-directional LSTM layer along with the attention layer and the employment of the additional node type vector. The node type vector enhanced the relation direction detection in our model. Regarding the slowest performing variant, the large embeddings size led to the reported speed in BERT's variant.

5 Conclusion

In this paper we explain that the relation prediction task has equal importance to the link prediction task. We also show that the usage of nodes' content information along with the structural information is important to have efficient graph representation. Accordingly, we propose a relation prediction model that utilizes a walks-based algorithm to retrieve the node structural information along with

the utilization of pre-trained language models to represent the textual content of the nodes. We argue that masked language models are costly in terms of computational resources and pre-trained language models can achieve equivalent results with effective exploitation. Our model does not require the nodes' description and shows competitive results when compared to the state of the art models in the same task.

References

1. Alqaaidi, S.K., Bozorgi, E., Kochut, K.J.: Multiple relations classification using imbalanced predictions adaptation. arXiv preprint arXiv:2309.13718 (2023)
2. 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)
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. Cui, Z., Kapanipathi, P., Talamadupula, K., Gao, T., Ji, Q.: Type-augmented relation prediction in knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 7151–7159 (2021)
6. Daza, D., Cochez, M., Groth, P.: Inductive entity representations from text via link prediction. In: Proceedings of the Web Conference 2021. pp. 798–808 (2021)
7. Demir, C., Moussallem, D., Ngomo, A.C.N.: A shallow neural model for relation prediction. In: 2021 IEEE 15th International Conference on Semantic Computing (ICSC). pp. 179–182. IEEE (2021)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
10. Han, J., Moraga, C.: The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International workshop on artificial neural networks. pp. 195–201. Springer (1995)
11. He, J., Jia, L., Wang, L., Li, X., Xu, X.: Mocosa: Momentum contrast for knowledge graph completion with structure-augmented pre-trained language models. arXiv preprint arXiv:2308.08204 (2023)
12. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
13. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. Advances in neural information processing systems **33**, 22118–22133 (2020)
14. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. arXiv preprint arXiv:1506.00379 (2015)

15. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
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. Sak, H., Senior, A., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128 (2014)
22. Shen, J., Wang, C., Gong, L., Song, D.: Joint language semantic and structure embedding for knowledge graph completion. arXiv preprint arXiv:2209.08721 (2022)
23. Shen, T., Zhang, F., Cheng, J.: A comprehensive overview of knowledge graph completion. Knowledge-Based Systems p. 109597 (2022)
24. Sheu, H.S., Li, S.: Context-aware graph embedding for session-based news recommendation. In: Proceedings of the 14th ACM Conference on Recommender Systems. pp. 657–662 (2020)
25. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
26. Tagawa, Y., Taniguchi, M., Miura, Y., Taniguchi, T., Ohkuma, T., Yamamoto, T., Nemoto, K.: Relation prediction for unseen-entities using entity-word graphs. In: Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13). pp. 11–16 (2019)
27. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv. org/abs/2307.09288 (2023)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
29. Wang, B., Shen, T., Long, G., Zhou, T., Wang, Y., Chang, Y.: Structure-augmented text representation learning for efficient knowledge graph completion. In: Proceedings of the Web Conference 2021. pp. 1737–1748 (2021)
30. Wang, L., Zhao, W., Wei, Z., Liu, J.: Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. arXiv preprint arXiv:2203.02167 (2022)
31. Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J.: Kepler: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics **9**, 176–194 (2021)
32. Xu, J., Chen, K., Qiu, X., Huang, X.: Knowledge graph representation with jointly structural and textual encoding. arXiv preprint arXiv:1611.08661 (2016)
33. Yani, M., Krisnadhi, A.A.: Challenges, techniques, and trends of simple knowledge graph question answering: a survey. Information **12**(7), 271 (2021)

34. Yao, L., Mao, C., Luo, Y.: Kg-bert: Bert for knowledge graph completion. arXiv preprint arXiv:1909.03193 (2019)
35. Youn, J., Tagkopoulos, I.: Kglm: Integrating knowledge graph structure in language models for link prediction. arXiv preprint arXiv:2211.02744 (2022)
36. Zha, H., Chen, Z., Yan, X.: Inductive relation prediction by bert. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 5923–5931 (2022)
37. Zhuang, Z., Liang, Z., Rao, Y., Xie, H., Wang, F.L.: Out-of-vocabulary word embedding learning based on reading comprehension mechanism. Natural Language Processing Journal **5**, 100038 (2023)