Enhancing Knowledge Graph Completion with Entity Neighborhood and Relation Context

Jianfang Chen[1], Kai Zhang[1(✉)], Aoran Gan[1], Shiwei Tong[2], Shuanghong Shen[3], and Qi Liu[1]

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China
{chenjianfang,gar}@mail.ustc.edu.cn, {kkzhang08,qiliuql}@ustc.edu.cn
[2] Tencent Company, Shenzhen, China
shiweitong@tencent.com
[3] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
shshen@iai.ustc.edu.cn

Abstract. Knowledge Graph Completion (KGC) aims to infer missing information in Knowledge Graphs (KGs) to address their inherent incompleteness. Traditional structure-based KGC methods, while effective, face significant computational demands and scalability challenges due to the need for dense embedding learning and scoring all entities in the KG for each prediction. Recent text-based approaches using language models like T5 and BERT have mitigated these issues by converting KG triples into text for reasoning. However, they often fail to fully utilize contextual information, focusing mainly on the neighborhood of the entity and neglecting the context of the relation. To address this issue, we propose KGC-ERC, a framework that integrates both types of context to enrich the input of generative language models and enhance their reasoning capabilities. Additionally, we introduce a sampling strategy to effectively select relevant context within input token constraints, which optimizes the utilization of contextual information and potentially improves model performance. Experiments on the Wikidata5M, Wiki27K, and FB15K-237-N datasets show that KGC-ERC outperforms or matches state-of-the-art baselines in predictive performance and scalability.

Keywords: Knowledge graph completion · Entity neighborhood · Relation context.

1 Introduction

Knowledge graphs (KGs) are structured collections of factual data represented as triples (h, r, t), where a head entity h is connected to a tail entity t through a relation r. Notable examples include Freebase [2], ConceptNet [15], and YAGO [16]. These KGs have been extensively utilized in various applications, ranging from recommendation systems and question answering to information retrieval. Their utility has further expanded to support complex AI tasks, including aspect-level

sentiment analysis [27] and adaptive learning prediction [10]. However, the inherent incompleteness of KGs often compromises the accuracy of downstream applications, highlighting the critical need for effective Knowledge Graph Completion (KGC) techniques.

Current KGC methods are broadly categorized into structure-based and text-based approaches. Structure-based methods map entities and relations into low-dimensional vectors and evaluate triple plausibility using scoring functions such as TransE [3], DistMult [23], ComplEx [20], and ROTATE [17]. These methods capture relational patterns effectively but face scalability challenges due to dense embedding learning and the need to score all entities for each prediction. In contrast, text-based methods like SimKGC [21], KGT5 [13], SKG-KGC [14], and KnowC [24] transform KG triples into text and use language models for reasoning, improving scalability. However, existing text-based KGC methods often neglect the full utilization of contextual information, focusing mainly on the *entity neighborhood* while ignoring the *relation context*. These contexts are crucial for language model inference.

To address this limitation, we introduce KGC-ERC, a novel framework that incorporates both entity neighborhood and relation context as ancillary cues within generative language models. When predicting a specific triple, we extract the surrounding entities and triples involving the given relation as context. These contextual information are merged with the target triple and input into the language model, enriching the entities and relations with additional relevant information and improving inference accuracy. However, due to the input token limitations of language models, not all contextual information can be accommodated. To solve this, we propose a sampling strategy for both entity neighborhood and relation context. For entity neighborhood, we collect direct neighbors with diverse relationships to gather a broader range of entity attributes. For relation context, we classify relations into four types (1-n, n-1, n-n, 1-1) based on cardinality and design distinct sampling strategies for each type to ensure the collected context is useful.

We evaluate KGC-ERC on Wikidata5M [22], Wiki27K [11], and FB15K-237-N [11] datasets, comparing MRR and Hits@k metrics against state-of-the-art methods. Results show that KGC-ERC achieves superior performance. In summary, our contributions include:

- We introduce a novel framework that integrates entity neighborhood and relation context into the input of a generative language model, enhancing its inferential capabilities.
- We propose a sampling method for efficiently selecting entity neighborhood and extracting relevant relation context, ensuring that our model can process and utilize the most pertinent information within the input capacity constraints.
- KGC-ERC is evaluated on three benchmark datasets: Wikidata5M, Wiki27K, and FB15K-237-N. The results show that our model outperforms existing baselines in predictive performance and scalability.

2 Preliminaries

Knowledge Graph Completion Knowledge graphs (KGs) are structured representations of facts about entities and their relations, formally defined as a set of triples $\mathcal{G} = \{(h, r, t)\}$, where $h \in \mathcal{E}$ is the head entity, $r \in \mathcal{R}$ is the relation, and $t \in \mathcal{E}$ is the tail entity. Here, $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively. Knowledge Graph Completion (KGC) aims to predict missing triples in a KG using existing information. For incomplete triples such as $(h, r, ?)$, the goal is to identify the most probable entity to complete the triple.

Sequence-to-sequence KGC Seq2seq KGC treats the task as a text generation problem using generative language models. Instead of ranking all possible entities, the model directly generates the missing entity as a sequence. For example, the incomplete triple $(StylipS, genre, ?)$ is transformed into a textual query q: "$query : StylipS | genre$". To enrich the input, context from *Entity Neighborhood* (E_n) and *Relation Context* (R_c) is appended to form:

$$q' = concat(q, E_n(q), R_c(q)) \tag{1}$$

This enriched input q' is then passed to the model to generate the output tokens:

$$T_{out} = Decoder(Encoder(q')) \tag{2}$$

The goal is to generate the missing entity. Following the methodology of KGT5-context [9], we create an inverse triple (t, r^{-1}, h) for each original triple (h, r, t), where r^{-1} denotes the reciprocal relation. This approach focuses on predicting the tail entity while considering bidirectional relations.

3 KGC-ERC Framework

KGC-ERC enhances KGT5-context [9] by adding relation context to the query, enriching the input sequence and expanding contextual information beyond the entity neighborhood. Given input token limits, a *Selector* module samples relevant entity and relation context, filtering out less important data. As illustrated in Fig. 1, it retrieves entity and relation context for a triple with a missing entity, selects relevant context, appends it to the query, and feeds the augmented query into the language model for prediction.

3.1 Selector

The *Selector* module is a critical component of the KGC-ERC model, managing the vast amount of contextual information in the knowledge graph. It ensures the generative language model receives only the most relevant context through Entity Neighborhood Sampling and Relation Context Sampling. Inspired by the interactive attention mechanism proposed by Zhang et al. [28], this dual interaction enhances the model's understanding of the query and captures complex relationships within the knowledge graph.
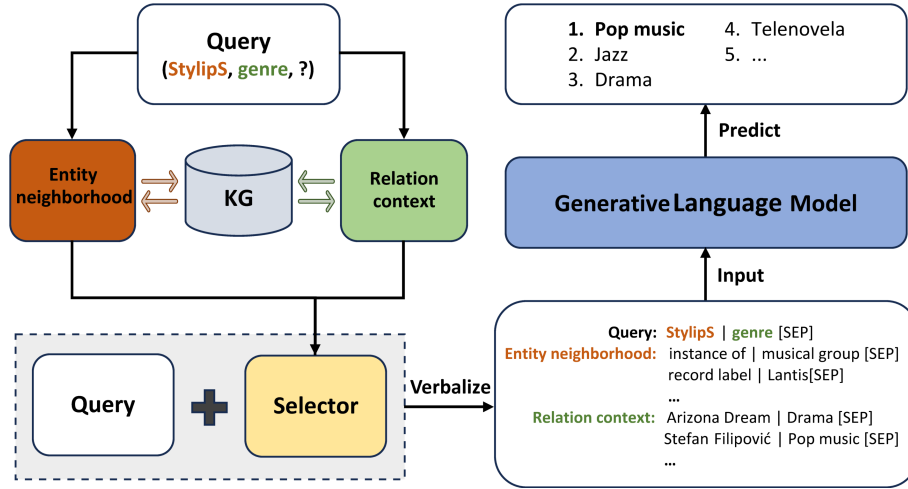
Fig. 1. Illustration of the KGC-ERC framework for knowledge graph completion. Real example from Wikidata5M dataset, best viewed in color.

Entity Neighborhood Sampling. Entity neighborhood sampling identifies all direct neighbors of the target entity, forming a set of relation-entity pairs (r, e). These pairs are grouped by relation and sorted by the size of each group in descending order. We then randomly select one pair from each group, up to a maximum of 50 pairs. This strategy ensures diverse attribute coverage and richer semantic representation. For example, for the entity *LeBron James* with neighbors (*child, Bryce Maximus James*), (*child, Bronny James*), and (*occupation, Basketball player*), selecting pairs from both *child* and *occupation* relations provides a more comprehensive representation than selecting only from *child* relations.

Relation Context Sampling. For relation context sampling, we focus on triples containing the same relation as the query. For 1-n relations, we prioritize triples involving both the head entity h and the relation r from the query. For n-1 relations, we ensure diversity in the tail entities to avoid repetition. For n-n relations, we combine the strategies of the previous two cases, selecting half of the triples based on each approach. For 1-1 relations, random sampling is employed. The maximum number of selected triples is capped at 50.

It is important to note that the information derived from the above two sampling processes explicitly excludes the target triple itself, as failure to do so would render the training task insignificant.

3.2 Verbalization

To construct the input for the generative language model, we concatenate the query with the entity neighborhood and the relation context selected by the

Table 1. Statistics of the datasets used in this paper.

	Entities	Relations	Train	Valid	Test
FB15K-237-N	13,104	93	87,282	7,041	8,226
Wiki27K	27,122	62	74,793	10,121	10,122
Wikidata5M	4,594,485	822	20,614,279	5,163	5,133

Selector (§ 3.1). The verbalization process transforms the query and contextual information into a coherent text sequence formatted as follows: "query: <query head entity> | <query relation> <SEP> entity neighborhood: <neighborhood 1 relation> | <neighborhood 1 tail entity> <SEP> ... <SEP> relation context: <context 1 head entity> | <context 1 tail entity> <SEP> ..."[4]. A real-world example is given in Fig. 1. The final combined input is truncated to a maximum of 512 tokens to fit the input requirements of the language model.

3.3 Training and Inference

Training involves constructing $(input, output)$ sequences from training triples (h, r, t), where inputs are queries and context in text format, and outputs are missing entities. We use teacher forcing and cross-entropy loss, without explicit negative sampling. Instead, each token is assessed step-by-step against all possibilities, simplifying the process.

For inference, given a query q, we verbalize q and its context, then input it into the model. The generative language model generates candidate entities via autoregressive decoding. This process is repeated multiple times to generate a set of candidate entities. Candidate sequences are scored using the method from KGT5 [13], maximizing the likelihood of correct generation. We discard outputs that do not map to valid entity mentions and remove any duplicate entities from the candidate set to ensure uniqueness. This ensures that only a subset of potential entities (far fewer than the total number in $\mathcal{E}$) need to be scored and ranked. If the correct entity is missing from the candidate set, it is assigned a score of $-\infty$, allowing the use of conventional KGC evaluation metrics.

4 Experiments

4.1 Experiment Setup

Datasets. We evaluate KGC-ERC on FB15K-237-N [11], Wiki27K [11], and Wikidata5M [22] datasets. Detailed statistics are provided in Table 1. FB15K-237-N, derived from FB15K-237 [19], contains approximately 13k entities and 93 relations. Wiki27K, constructed from Wikidata, includes around 27k entities and 62 relations. In contrast, Wikidata5M, combining Wikidata with Wikipedia pages, encompasses nearly 5 million entities and 20 million triples.

---

[4] When entity descriptions are accessible, we incorporate them by adding "description: <description of query head entity>"

Table 2. Experiment results on Wikidata5M dataset. Results of [†] are from KGT5 [13] and the other results are from the corresponding papers. The first group omits textual information, the second group employs mention names, the third group additionally entity descriptions. Best per group underlined, best overall bold.

Model	MRR	H@1	H@3	H@10	Params
Structure-based Methods					
TransE [3] †	0.253	0.170	0.311	0.392	2.4B
DistMult [23] †	0.253	0.209	0.278	0.334	2.4B
SimplE [7] †	0.296	0.252	0.317	0.377	2.4B
RotatE [17] †	0.290	0.234	<u>0.322</u>	0.390	2.4B
QuatE [29] †	0.276	0.227	0.301	0.359	2.4B
ComplEx [20] †	<u>0.308</u>	<u>0.255</u>	-	<u>0.398</u>	<u>614M</u>
Text-based Methods					
KEPLER [22]	0.210	0.173	0.224	0.277	125M
MLMLM [4]	0.223	0.201	0.232	0.264	355M
KGT5 [13]	0.300	0.267	0.318	0.365	**60M**
ReSKGC [26]	0.363	0.334	0.386	0.416	**60M**
KGT5-context [9]	0.378	0.350	0.396	0.427	**60M**
KGC-ERC (ours)	<u>0.386</u>	<u>0.360</u>	<u>0.403</u>	<u>0.436</u>	**60M**
Text-based Methods With Descriptions					
SimKGC + Desc. [21]	0.358	0.313	0.376	0.441	220M
SKG-KGC + Desc. [14]	0.366	0.323	0.382	0.446	220M
KGT5-context + Desc. [9]	0.426	0.406	0.440	0.460	**60M**
KnowC + Desc. [24]	0.426	0.373	0.447	**<u>0.525</u>**	220M
KGC-ERC + Desc. (ours)	**<u>0.433</u>**	**<u>0.412</u>**	**<u>0.448</u>**	0.467	**<u>60M</u>**

Evaluation Metrics. We evaluate KGC-ERC using the entity ranking task. For each test triple (h, r, t), the model predicts the tail entity t given the head entity h and relation r. Head entity prediction is achieved by reversing the triple. We sample from the decoder, discarding invalid entity mentions.

 We use four metrics: mean reciprocal rank (MRR) and Hits@k (H@k) for $k \in \{1, 3, 10\}$. MRR measures the mean of the reciprocal ranks of the correct entities, while H@k indicates the proportion of correct predictions within the top-k results. In case of ties, we use the mean rank to avoid misleading results [18]. Both MRR and H@k are calculated under the filtered setting [3], which excludes known true triples from the training, validation, and test sets. We report the average of these metrics for both head and tail entity predictions.

Implementation Details. We follow the settings of KGT5-context [9] and use T5 [12] models for our experiments. For the large-scale Wikidata5M dataset, we train the T5-small model from scratch due to the substantial amount of data. For the smaller FB15K-237-N and Wiki27K datasets, we fine-tune the T5-base model. All models use the SentencePiece tokenizer [12]. Details are as follows:

– **Wikidata5M**: Trained on 4 Nvidia 4090 GPUs with a batch size of 64 (effective batch size 256). The optimizer is AdaFactor, and training lasts for

6 epochs. Contextual information (entity neighborhood and relation context) is sampled up to 50 instances each.

– **FB15K-237-N and Wiki27K**: Trained on a single Nvidia A100 GPU with a batch size of 16. The optimizer is AdamW with a learning rate of 5×10^{-5} after testing multiple values. Training lasts for 10 epochs. Contextual information is sampled up to 10 entity neighbors and 50 relation contexts.

The sampling sizes for entity neighbors and relation contexts were determined through extensive experiments to optimize model performance, ensuring that the chosen numbers are data-driven rather than arbitrary. For inference, we sample 500 entities from the decoder across all datasets.

4.2 Experiment Results

Baselines. We compare our KGC-ERC model with various baselines across the three datasets. For Wikidata5M, we include structure-based methods such as TransE, DistMult, SimplE, RotatE, QuatE, and ComplEx, as well as text-based methods like KEPLER, MLMLM, KGT5, ReSKGC, KGT5-context, SimKGC, SKG-KGC, and KnowC. Notably, KGT5, ReSKGC, KGT5-context, and KGC-ERC utilize generative language models. For Wiki27K and FB15K-237-N, we compare with structure-based methods (TransE, ConvE, TuckER, RotatE) and text-based models (KG-BERT, LP-RP-RR, PKGC, VKGC), where PKGC and VKGC incorporate entity descriptions.

Main Results. As shown in Table 2, our KGC-ERC method achieves superior performance across the Wikidata5M, Wiki27K, and FB15K-237-N datasets, outperforming existing baselines in nearly all evaluation metrics. On Wikidata5M, KGC-ERC achieves an MRR of 0.386 and H@1 of 0.360, outperforming KGT5-context and ReSKGC. When incorporating entity descriptions, KGC-ERC further improves to an MRR of 0.433 and H@1 of 0.412, outperforming the best baseline, KnowC, in all but H@10. Importantly, KGC-ERC achieves these results with only 60M parameters, demonstrating its efficiency compared to traditional structure-based methods that often require billions of parameters.

In Table 3, which reports results on Wiki27K and FB15K-237-N, KGC-ERC again surpasses both structure-based and text-based methods, achieving the highest MRR, H@1 and H@3. Specifically, our method achieves an MRR of 0.305 and 0.343 on Wiki27K and FB15K-237-N, respectively. This shows its robust performance on different benchmarks.

4.3 Ablation Study

To evaluate the impact of key components in the KGC-ERC model, we conducted ablation studies focusing on the relation context and the customized sampling strategy. Specifically, we tested the model without relation context (using only entity mentions, relation mentions, and 1-hop neighborhood information) and with a random sampling strategy instead of the customized one.

Table 3. Experiment results on Wiki27K and FB15K-237-N datasets. Results of [†] are from PKGC [11] and the other results are from the corresponding papers. The best score is in bold.

Model	Wiki27K				FB15K-237-N			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Structure-based Methods								
TransE [3] †	0.155	0.032	0.228	0.378	0.255	0.152	0.301	0.459
ConvE [5] †	0.226	0.164	0.244	0.354	0.273	0.192	0.305	0.429
TuckER [1] †	0.249	0.185	0.269	0.385	0.309	0.227	0.340	0.474
RotatE [17] †	0.216	0.123	0.256	0.394	0.279	0.177	0.320	0.481
Text-based Methods								
KG-BERT [25] †	0.192	0.119	0.219	0.352	0.203	0.139	0.201	0.403
LP-RP-RR [8] †	0.217	0.138	0.235	0.379	0.248	0.155	0.256	0.436
PKGC [11]	0.285	0.230	0.305	0.409	0.332	0.261	0.346	0.487
VKGC [6]	0.296	0.241	0.320	**0.422**	0.338	0.266	0.369	**0.489**
KGC-ERC (ours)	**0.305**	**0.267**	**0.322**	0.381	**0.343**	**0.280**	**0.375**	0.465

Table 4. The ablation results on the Wikidata5M dataset.

Model	MRR	H@1	H@3	H@10
w/o relation context	0.375	0.348	0.392	0.425
w/o sampling strategy	0.382	0.356	0.398	0.431
KGC-ERC	**0.386**	**0.360**	**0.403**	**0.436**

The results (Table 4) show that excluding the relation context reduces the MRR from 0.386 to 0.375 and H@1 from 0.360 to 0.348. Similarly, replacing the customized sampling strategy with random sampling degrades performance, with the MRR falling to 0.382 and H@1 to 0.356. These findings highlight the importance of both components in enhancing the model's reasoning abilities and optimizing the use of the limited input window. The integration of relation context and the tailored sampling strategy are crucial for achieving state-of-the-art performance and ensuring robust generalizability.

5 Conclusion

We introduced KGC-ERC, a novel framework for knowledge graph completion that integrates entity neighborhood and relation context into a generative language model. Our approach achieved superior performance across multiple datasets, outperforming existing baselines in most metrics. Ablation studies demonstrated the importance of relation context and a tailored sampling strategy in enhancing model effectiveness and optimizing the use of the input window. Future work will focus on developing adaptive sampling techniques to dynamically select context more aligned with the query, thereby improving predictive accuracy and broadening the model's applicability.

References

1. Balažević, I., Allen, C., Hospedales, T.: Tucker: Tensor factorization for knowledge graph completion. 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. 5185–5194 (2019)
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. Clouatre, L., Trempe, P., Zouaq, A., Chandar, S.: Mlmlm: Link prediction with mean likelihood masked language model. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 4321–4331 (2021)
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018. vol. 32, pp. 1811–1818. AAI Publications (2018)
6. Han, S., Guan, Z., Li, S., Wang, J., Zhou, X.: Knowledge graph completing with dual confrontation learning model based on variational information bottleneck method. In: 2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security (QRS). pp. 741–750. IEEE (2023)
7. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. Advances in neural information processing systems **31** (2018)
8. Kim, B., Hong, T., Ko, Y., Seo, J.: Multi-task learning for knowledge graph completion with pre-trained language models. In: Proceedings of the 28th international conference on computational linguistics. pp. 1737–1743 (2020)
9. Kochsiek, A., Saxena, A., Nair, I., Gemulla, R.: Friendly neighbors: Contextualized sequence-to-sequence link prediction. In: Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023). pp. 131–138 (2023)
10. Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G.: Ekt: Exercise-aware knowledge tracing for student performance prediction. IEEE Transactions on Knowledge and Data Engineering **33**(1), 100–115 (2019)
11. Lv, X., Lin, Y., Cao, Y., Hou, L., Li, J., Liu, Z., Li, P., Zhou, J.: Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. In: Findings of the Association for Computational Linguistics: ACL 2022. pp. 3570–3581 (2022)
12. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
13. Saxena, A., Kochsiek, A., Gemulla, R.: Sequence-to-sequence knowledge graph completion and question answering. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2814–2828 (2022)

14. Shan, Y., Zhou, J., Peng, J., Zhou, X., Yin, J., Wang, X.: Multi-level shared knowledge guided learning for knowledge graph completion. Transactions of the Association for Computational Linguistics **12**, 1027–1042 (2024)

15. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31 (2017)

16. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706 (2007)

17. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HkgEQnRqYQ

18. Sun, Z., Vashishth, S., Sanyal, S., Talukdar, P., Yang, Y.: A re-evaluation of knowledge graph completion methods. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 5516–5522 (2020)

19. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality. pp. 57–66 (2015)

20. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International conference on machine learning. pp. 2071–2080. PMLR (2016)

21. Wang, L., Zhao, W., Wei, Z., Liu, J.: Simkgc: Simple contrastive knowledge graph completion with pre-trained language models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4281–4294 (2022)

22. 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)

23. Yang, B., Yih, S.W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR) 2015 (2015)

24. Yang, G., Liu, Y., Zhang, L., Zhang, L., Xie, H., Mao, Z.: Knowledge context modeling with pre-trained language models for contrastive knowledge graph completion. In: Findings of the Association for Computational Linguistics ACL 2024. pp. 8619–8630 (2024)

25. Yao, L., Mao, C., Luo, Y.: Kg-bert: Bert for knowledge graph completion. arXiv preprint arXiv:1909.03193 (2019)

26. Yu, D., Yang, Y.: Retrieval-enhanced generative model for large-scale knowledge graph completion. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2334–2338 (2023)

27. Zhang, K., Liu, Q., Qian, H., Xiang, B., Cui, Q., Zhou, J., Chen, E.: Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **35**(1), 377–389 (2021)

28. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5773–5780 (2019)

29. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. Advances in neural information processing systems **32** (2019)