mathematics

MDPI

Article

Generative Transformer with Knowledge-Guided Decoding for Academic Knowledge Graph Completion

Xiangwen Liu, Shengyu Mao, Xiaohan Wang and Jiajun Bu *

College of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China
* Correspondence: bjj@zju.edu.cn

Abstract: Academic knowledge graphs are essential resources and can be beneficial in widespread real-world applications. Most of the existing academic knowledge graphs are far from completion; thus, knowledge graph completion—the task of extending a knowledge graph with missing entities and relations—attracts many researchers. Most existing methods utilize low-dimensional embeddings to represent entities and relations and follow the discrimination paradigm for link prediction. However, discrimination approaches may suffer from the scaling issue during inference with large-scale academic knowledge graphs. In this paper, we propose a novel approach of a generative transformer with knowledge-guided decoding for academic knowledge graph completion. Specifically, we introduce generative academic knowledge graph pre-training with a transformer. Then, we propose knowledge-guided decoding, which leverages relevant knowledge in the training corpus as guidance for help. We conducted experiments on benchmark datasets for knowledge graph completion. The experimental results show that the proposed approach can achieve performance gains of 30 units of the MRR score over the baselines on the academic knowledge graph AIDA.

Keywords: knowledge graph; transformer; generation

MSC: 68T50; 68T30

Citation: Liu, X.; Mao, S.; Wang, X.; Bu, J. Generative Transformer with Knowledge-Guided Decoding for Academic Knowledge Graph Completion. *Mathematics* **2023**, *11*, 1073. https://doi.org/10.3390/math11051073

Academic Editors: Tianxing Wu, Yuxiang Wang and Ningyu Zhang

Received: 8 January 2023
Revised: 5 February 2023
Accepted: 13 February 2023
Published: 21 February 2023

1. Introduction

Knowledge graphs (KGs), also known as semantic networks, can represent much symbolic knowledge, including entities, relations, and events; thus, they appeal to many researchers. With the fast development of artificial intelligence (AI), knowledge graphs can provide back-end support for widespread tasks and be applied to real-world applications, including search engines [1,2], recommendation systems [3], medical health [4], natural language understanding [5–8], commonsense reasoning [9], time-serious prediction [10], and cross-discipline scenarios [11]. Note that most of the existing knowledge graphs are far from completion [12]; thus, knowledge graph completion—the task of extending a knowledge graph with missing entities and relations—attracts many researchers.

In the academic domain, knowledge graphs are essential resources due to their potential value for scientific research. With the fast development of knowledge graphs, we have witnessed many popular knowledge graphs, including OpenCitations [13], Core [14], Microsoft Academic Graph [15], Aminer [16], Open Research Knowledge Graph [17], and so on. These resources provide essential advantages for various research studies and can help research policymakers and funding agencies. Moreover, academic knowledge graphs can provide support for scientific literature research, can recommend related papers and authors, and can help discover future research trends. Nevertheless, with the fast evolution of the academic domain, most of those knowledge graphs are far from complete, which is a major obstacle for their application. For example, as shown in Figure 1, the relation type between “author: Yann Lecun” and “paper: Convolutional Networks and Applications in Vision” is missing, which makes the relational triple less informative and even ambiguous.

Therefore, it is crucial for models to be able to complete missing entities/relations for academic knowledge graphs.

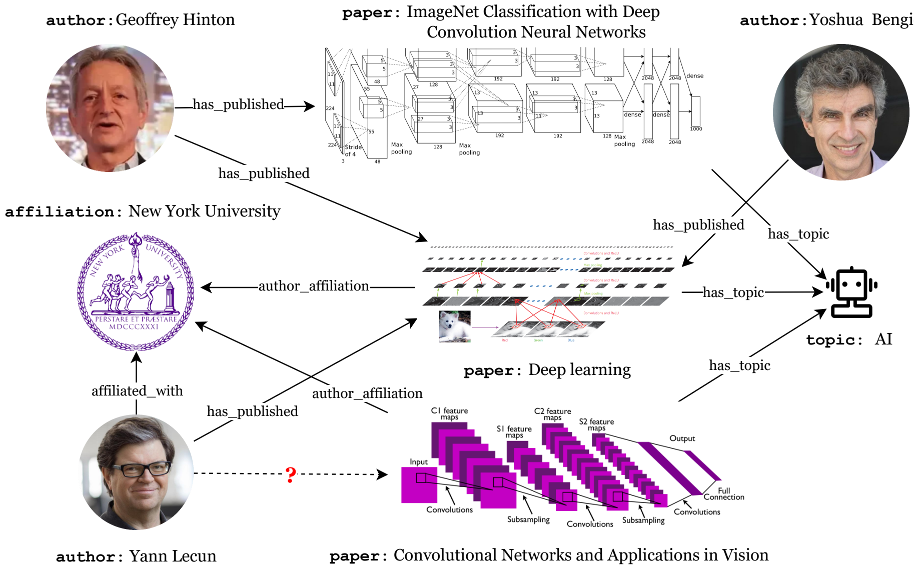


Figure 1. Academic knowledge graph completion. This paper focuses on addressing this task through link prediction—for example, inferring the relation between "author: Yann Lecun" and "paper: Convolutional Networks and Applications in Vision".

Conventionally, information extraction technologies [18–21] such as named entity recognition [22], relation extraction [23], and event extraction [24,25], can extract knowledge from a text corpus and help to complete the missing academic knowledge. However, when facing large-scale real-world academic knowledge graphs, it is necessary to develop efficient knowledge graph completion methodologies. Concretely, many knowledge-graph-embedding approaches (e.g., TransE [26], RotatE [27]) that utilize low-dimensional representations to embed entities and relations, which are dubbed knowledge graph embeddings, and obtain target predictions via score functions for those embeddings have been developed. More recently, some approaches (e.g., KG-BERT [28]) have tried to leverage pre-trained language models in natural language processing and encode relational triples with transformers to obtain the final score for a missing target. Note that most of those methodologies followed the discrimination paradigm with a pre-defined scoring function for knowledge graph representation learning. We argue that these discrimination approaches are time-consuming due to the costly scoring of candidate relational triples when inferring missing knowledge. Moreover, discrimination approaches lack rich interactions between relations and entities and suffer from the issue of the instability of negative sampling.

In this paper, to address the above-mentioned challenges, we propose a new technical solution for academic knowledge graph completion, which is named a generative transformer with knowledge-guided decoding (GTK). Specifically, we introduce generative academic knowledge graph pre-training with a transformer. We formulate knowledge graph completion as a sequence-to-sequence task and leverage BART-style [29] pre-training. We further propose knowledge-aware demonstrations inspired by GPT-3 [30], which concatenate some selected samples to the input sequence. Since there exist rich semantics and types of constraints for knowledge graph completion, we propose knowledge-guided decoding, which leverages relevant knowledge in the training corpus as guidance for help. We further utilize a policy gradient [31] to learn the generation strategy for maximizing the correctness of the generated knowledge and mitigating exposure bias.

We evaluate the proposed model **GTK** against previous baselines, including TransE [26] and KG-BERT [28], on the benchmark datasets of AIDA, MAG, and other popular knowledge graph completion datasets, namely, FB15K-237, WN18RR, and OpenBG500 [32,33].

The extensive experimental results illustrate that the proposed approach can yield better performance than that of the baselines with a faster inference speed. In total, we summarize the contributions of this work as follows:

- We propose a novel model of a generative transformer with knowledge-guided decoding (GTK) for academic knowledge graph completion.
- We propose knowledge-aware demonstration and knowledge-guided decoding for knowledge graph completion.
- We evaluate the model on various benchmark datasets for knowledge graph completion, which demonstrates the effectiveness of the proposed approach.

In the following sections, we will introduce the related work (Section 2) and background (Section 3). Then, we will introduce the technical details of the proposed approach, including link prediction as Seq2Seq generation (Section 4.1), knowledge-aware demonstration (Section 4.2), and knowledge-guided decoding (Section 4.3). Finally, we will introduce the experiments (Section 5) and conclude the paper (Section 6).

2. Related Work

Constructing knowledge graphs from scratch is intractable and requires the efficient completion of missing triples. Given the trait of incompleteness, knowledge graph completion (KGC) approaches can be divided into embedding-based methods and text-based methods.

In preliminary works, embedding-based methods showed many advantages in handling KGC tasks [34]. By converting triples into continuous embedding vectors, relations between entities are treated as special mapping functions that score pairs of top-k entities in a low-dimensional space. Such methods include TransE [26], TransR [35], and RotatE [27]. The Trans-series models, which are translation based, are equipped with both simplicity and strong interpretability in practice. Another type of embedding model, a semantic matching model, utilizes semantic similarity functions to capture the plausibility of latent triples in knowledge graphs. These methods emphasize the structural information observed in triples rather than contextualized information, and they include DistMult [36], ComplEx [37], Tucker [38], and Trans4E [39], which is an embedding model fit for academic knowledge graphs with N to M relations.

In contrast, text-based methods attempt to incorporate available texts for representation by leveraging state-of-the-art pre-trained language models [28,40–42]. When the architecture of a transformer occurs, it is employed in CoKE [43] to encode the path and edge sequences in graphs. Before long, pre-trained language models advanced contextualized text representation learning. KG-BERT [28], which took the first step in applying BERT [44] for KGC, concatenated the text of triples as sequences and handled the sequence classification problem with binary cross-entropy optimization. Referring to KG-BERT, StAR [45] followed the textual encoding paradigm and applied a Siamese-style textual encoder that partitioned each triplet into two asymmetric parts. kNN-KGE [46] derived knowledge graph embeddings by interpolating the entity distribution linearly from the knowledge store with the k-nearest neighbors (kNN) model. Apart from switching to the InfoNCE loss, SimKGC [47] introduced three types of negatives: pre-batch negatives, self-negatives, and in-batch negatives. LMKE [48] formulated a contrastive learning framework with the goal of augmenting long-tail entity representations with the inductive capabilities of description-based approaches. Motivated by prompt-based models, PKGC [49] manually defined each triple as a natural prompt sentence and further introduced soft prompts for better representation. Considering both the structure of a knowledge graph and the underlying textual description, StATIK [50] used language models to extract semantic information while incorporating a structure via message-passing neural networks. LASS [51], a joint embedding approach, embedded knowledge graphs via fine-tuned BERT or RoBERTa with respect to a probabilistic structured loss.

Rather than using discriminative methods, a line of recent work changed KGC into sequence-to-sequence generation problems. Posing link prediction as a sequence-to-sequence task, KGT5 [52] adopted a transformer model and was further fine-tuned; it used autoregressive decoding to reduce the model size by up to 98% in comparison with that of KGE models. For better representation learning and fast inference, GenKGC [33] proposed entity-aware hierarchical decoding to generate In addition, KG-S2S [53], which uniformly codified the representation of KG facts into "flat" text, was capable of handling a variety of verbalizable graph structures.target entities.

However, most previous works regarded entities and relations as plain text, ignoring the structural bias when generating target knowledge, which hindered their knowledge graph completion performance.

3. Background

A knowledge graph represents the fact that knowledge exists in the form of a triple. In this paper, we target the knowledge graph completion task with the aim of predicting missing parts (entities or relations) based on existing triples.

A knowledge graph $G = (\mathcal{E}, \mathcal{R})$ composed of an entity set $\mathcal{E} = \{e_1, e_2, e_3, \ldots\}$ and an edge set $\mathcal{R} = \{r_1, r_2, r_3, \ldots\}$ is given. The node set V consists of both entities and relations, that is, $V = \mathcal{E} \cup \mathcal{R}$. The graph structure can be represented as an adjacency matrix $\mathbf{A} \in \{0,1\}^{|V| \times |V|}$, where $|V|$ denotes the number of nodes in the graph. The element $\mathbf{A}[i,j]$ in the adjacency matrix equals 1 if there exists a link between nodes v_i and v_j; otherwise, $\mathbf{A}[i,j] = 0$, where i and j refer to the ID of the node v.

For the knowledge graph completion task, a triple is denoted as $(v_{src}, v_p, v_{target})$, or $\mathcal{T}$ for short. We define the contextualized sub-graph $\mathcal{T}_G$ as the subgraph surrounding the triple $\mathcal{T}$ (including the triple itself). Meanwhile, the corresponding graph structure of the contextualized sub-graph sequence is preserved in matrix $\mathbf{A}_G$. $\mathcal{T}_M$ denotes the masked contextualized sub-graph. Therefore, the goal of knowledge graph completion is to learn a mapping $f : \mathcal{T}_M, \mathbf{A}_G \rightarrow \mathbf{Y}$, where $\mathbf{Y} \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{R}|}$ are the label sets of the triple $\mathcal{T}$. f refers to the score function, which maps facts into embeddings to generate label sets.

3.1. Knowledge-Graph-Embedding-Based Methods

Knowledge-graph-embedding-based methods are the approaches for knowledge graph completion that have recently been dominant. Given a triple (h, r, t), the key is to find the relationships between relations and entities via a score function as follows:

$$score(h, r, t) = f_r(h, t) \tag{1}$$

The relevance or distance is calculated with a pre-defined score function f, and there are many score functions [26,27,35]. It should be noted that the translational distance method is essentially a ranking model, and it requires negative samples to maximize the difference between the positive and negative triples. A straightforward approach to getting negative triples (h', r, t) or (h, r, t') is randomly replacing head or tail entities. Thus, we can optimize knowledge-graph-embedding-based knowledge graph completion as follows:

$$\begin{aligned} L = &-\log\ \sigma(\gamma - score_{pos}(h, r, t)) \\ &- \sum \log\ \sigma(score_{neg}(h, r, t) - \gamma) \end{aligned} \tag{2}$$

where γ is a marginal hyper-parameter, and σ is a sigmoid function.

3.2. Pre-Trained Language-Model-Based Methods

With the fast development of pre-training, there are many approaches that leverage pre-trained language models for the knowledge graph completion task. For example, KG-BERT [28] utilizes a single encoder to encode relational triples in a knowledge graph via text description. Formally, we have the score of the relational triple as follows:

$$Score(h, r, t) = \mathrm{TransformerEnc}(X^h, X^r, X^t), \tag{3}$$

where TransformerEnc is the BERT [44] model followed by a binary classifier. Note that the previous studies mostly followed the discrimination paradigm; however, they may suffer from inefficiency and inflexibility issues.

There are several reasons for this: (1) A ranking model is not a straightforward way to predict the missing part of a triple, since it needs to repeatedly compute all scores of the candidates. (2) A score function is essentially an artificial constraint, and handcrafting the best-performing score function is like finding a needle in a haystack. (3) The negative sampling process is full of uncertainty with different batch sizes or sample difficulties. Thus, it is intuitive to develop a new solution for better knowledge graph completion, and the generation paradigm is a good fit for that role.

4. Approach

4.1. Link Prediction as Seq2Seq Generation

In this paper, we follow BART [29] in formulating knowledge graph completion as sequence-to-sequence generation. Specifically, we regard the entities and relations as token sequences and utilize an encoder–decoder architecture for generation. As shown in Figure 2, we follow [28] in leveraging text descriptions rather than specific embeddings to represent entities and relations. Given an input triple with a missing tail entity $(e_i, r_j, ?)$, we get the descriptions dr_j and d_{e_i} of r_j and e_i, respectively. Then, we concatenate these embeddings to obtain the input sequence of $< e_i, r >$. Note that we aim to generate the token sequence of e_k, which is similar to natural language generation. For example, given the text sequence regarding the Query <Steve Jobs, founded, ?>, we aim to generate a target entity with the text sequence *Apple Inc*. It is intuitive to pre-train the generation model to boost performance.

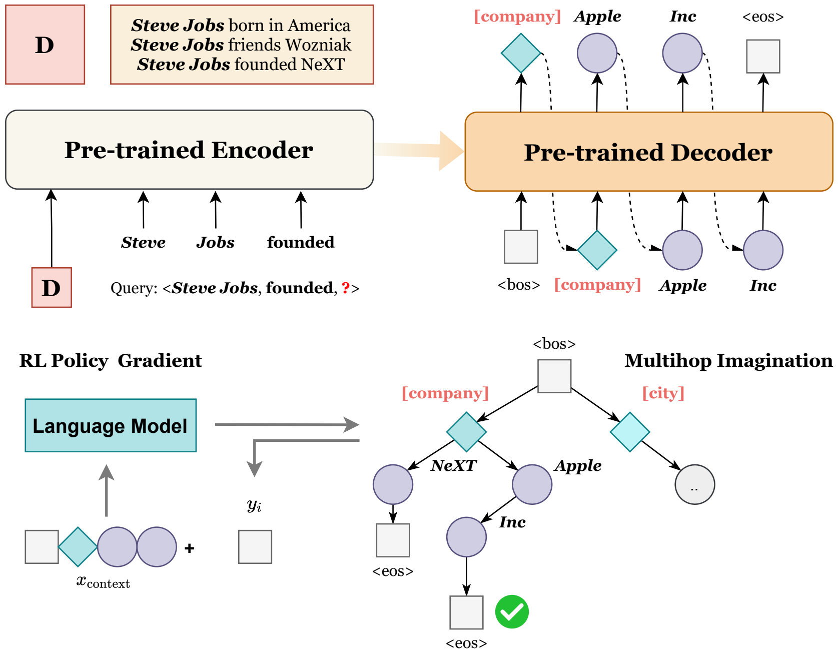


Figure 2. The architecture of the proposed model of the generative transformer with knowledge-guided decoding (GTK) for academic knowledge graph completion. We utilize a sequence-to-sequence-based architecture with a pre-trained transformer and leverage knowledge guidance in the input and during generation.

We leveraged a large-scale corpus (while filtering the triples, such as FB15K-237, during the evaluation to avoid data leakage) from Wikipedia and aligned it with the Wikidata knowledge graph for pre-training.We pre-trained the model with 16 A100 for one week. During inference, we used autoregressive generation, and more details can be found in Section 4.3.

4.2. Knowledge-Aware Demonstration

Inspired by GPT-3 [30], we propose the utilization of relevant triples in the training set as demonstrations to boost the generation performance. Regarding the long-tailed distribution in a large-scale knowledge graph, for example, only 37 instances of the relation `film/type_of_appearance` exist in the popular FB15k-237 dataset. We utilized the prior knowledge distribution of entities and relations as guidance for sampling demonstrations. To be specific, we first filtered the relations based on the input sequence and then sampled the triples according to the distribution of entity types. Thus, those long-tailed relational triples have more representative demonstrations for help. Finally, we can construct the demonstration examples $\{r_{\text{in}}, t_{\text{train}}\}$ and obtain the final input sequence as follows:

$$x = \texttt{<bos>}\ \text{demonstration}(r_j)\ \texttt{<sep>}\ d_{e_i}\ d_{r_j}\ \texttt{<sep>}$$

4.3. Knowledge-Guided Decoding

During inference, we can use the Beam Search to obtain the top k entities in $\mathcal{E}$ (k is the hyper-parameter of the beam size). Note that there is no native sampling, as we directly optimize via the generation of the right entities. Since academic knowledge graphs contain rich semantic and schematic knowledge, such as entity types, we propose knowledge-guided decoding to boost performance. Firstly, we sample the lowest-frequency entity types and construct a trie tree due to the fact that these low-frequency entities are challenging to decode. We add some special tokens as entity types in the vocabulary of the language models for knowledge-guided decoding. We first utilize the standard cross-entry loss to optimize the log-likelihood for generation as follows:

$$\mathcal{L} = -\log p_\theta(y \mid x) \tag{4}$$

Inspired by [54], we then formulate sequence generation as a reinforcement learning problem. The action is the current step's output token (i.e., $a_t = y_t$), and the state is the same as that of the tokens generated before t. Then, we utilize the policy π_t to pick token y_t (action a_t) given the state $s_t = y_{<t}$ through multi-hop imagination to constrain the output of generation. We use a policy gradient [31] to learn optimized strategies.

5. Experiments

5.1. Datasets

We followed [39] and use the AIDA+MAG knowledge graph for evaluation. It contains 180 K triples with 68,906 entities, including affiliated organizations, authors, publications, and so on. The subset of the hasGRIDType relation contains research paper entities. The subset of the hasTopic relation mainly consists of triples of articles with associated topics. The datasets were split according to a ratio of 8/1/1 (80% for the training set, 10% for the validation set, 10% for the test set) to construct the training, validation, and test sets.

We also evaluated the proposed approach on FB15k-237, WN18RR, and a real-world knowledge graph for e-commerce, OpenBG500 (OpenBG500 is a subset of the open business KG from https://kg.alibaba.com/ accessed on 1 September 2022). For FB15k-237, we leveraged descriptions from the Wikipedia page for entities and relations. For WN18RR, each entity contained a word sense, and we leveraged the word definitions for descriptions. For OpenBG500, we leveraged the descriptions from the e-commerce description page for entities and relations. We illustrate the statistical details of FB15k-237, WN18RR, and OpenBG500 in Table 1.

Table 1. Statistics of FB15k-237, WN18RR, and OpenBG500. # refers to the number.

Dataset	# Ent	# Rel	# Train	# Dev	# Test
AIDA	68,906	2	144,000	18,000	18,000
WN18RR	40,943	11	86,835	3034	3134
FB15k-237	14,541	237	272,115	17,535	20,466
OpenBG500	269,658	500	1,242,550	5000	5000

5.2. Metrics

We used the metrics of hits@1, hits@3, hits@10, and MRR for evaluation. We sorted the scores of the entities in the candidate set to obtain a ranked list. We obtained the hits at k metric (Hits@k) by counting the number of times that the correct triple appeared at position k. We defined $rank_{\text{head}}$ as the position of the correct entity in the rank list. We calculated the reciprocal rank with $1/rank_{\text{head}}$ and repeated the procedure by predicting the tail entity to obtain the reciprocal rank $1/rank_{\text{tail}}$. Additionally, the mean rank (MR) refers to the $rank_{\text{head}}$ and $rank_{\text{tail}}$ averaged across all triples in the KG. The mean reciprocal rank (MRR) refers to the mean of these two values. Formally, we have:

$$\text{MRR} = \frac{1}{2|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\frac{1}{rank_{\text{head}}} + \frac{1}{rank_{\text{tail}}} \right) \tag{5}$$

It is to be noted that the entities in the test triple were assumed to be seen in the training set.

5.3. Settings

We used 16 Nvidia A 100 GPUs for model pre-training and fine-tuning. We developed our model with Pytorch and utilized the BART-base as a backbone. The learning rate was set to 5×10^{-5}. We leveraged a grid with a validation set search to find the optimized hyper-parameters. We also utilized early stooping for each dataset. Specifically, we recorded the test-set performance of the best-performing model on the development set for each random seed and reported the final performance as the average of the results across the five seeds.

5.4. Results

On the one hand, from Table 2, we observed that on the subsets of hasTopic and hasGRIDType, the previous baselines TransE, RotatE, and QuatE were able to obtain better performance than that of ComplEx, while Trans4E was able to achieve better performance than all of them. We argue that Trans4E can better represent academic knowledge graphs with N to M relations, thus leading to better performance. We further found that the performance of knowledge-embedding models in the hasGRIDType subset was much higher than that in the hasTopic subset. We think that this was because the hasGRIDType subset is a relatively easy subset of academic knowledge graphs. We also noticed that the proposed GTK model was able to obtain the best performance on both academic knowledge graph datasets. On the other hand, note that previous knowledge-graph-embedding approaches regard relations and entities as dense vectors in the same space; thus, they struggle with the memory cost for large-scale academic knowledge graphs. However, our GTK has a fixed memory size with a pre-trained language model, which contains 110 M parameters, and will not scale with the size of entities.

Table 2. Evaluation results on the AIDA academic knowledge graph.

Model Type	**hasTopic**					**hasGRIDType**				
	MR	MRR	Hits@1	Hits@3	Hits@10	MR	MRR	Hits@1	Hits@3	Hits@10
TransE	3982	0.400	0.294	0.462	0.592	1	0.968	0.944	0.990	1.000
RotatE	4407	0.433	0.332	0.492	0.622	1	0.953	0.933	0.975	0.996
QuatE	1353	0.426	0.341	0.472	0.581	1	0.957	0.928	0.983	0.998
ComplEx	5855	0.099	0.077	0.109	0.129	1566	0.566	0.531	0.596	0.609
Trans4E	3904	0.426	0.318	0.492	0.628	1	0.968	0.944	0.995	0.998
GTK	3835	0.456	0.355	0.488	0.650	1	0.975	0.967	0.998	1

From Table 3, we also observe that **GTK** obtained better performance than that of KG-BERT [28] on WN18RR, FB15k-237, and OpenBG500. Moreover, the proposed model had a faster inference speed, since it directly generated the target entities rather than engaging in the costly scoring of candidate relational triples. For example, when the number of target candidate entities was very large, it cost lots of time (KG-BERT took about 91,100 s) for inference, as shown in Table 4. However, the proposed model only had to generate the top *k* entities with knowledge-guided decoding, which was faster.

Table 3. Evaluation results on WN18RR, FB15k-237, and OpenBG500. Note that the ◇ resulting numbers were reported by previous works, which means that we reproduced the experimental results on OpenBG500 and took other results from the original papers.

	WN18RR			**FB15k-237**			**OpenBG500**		
Method	**Hits@1**	**Hits@3**	**Hits@10**	**Hits@1**	**Hits@3**	**Hits@10**	**Hits@1**	**Hits@3**	**Hits@10**
				Graph-embedding approach					
TransE [26] ◇	0.043	0.441	0.532	0.198	0.376	0.441	0.207	0.340	0.513
DistMult [36] ◇	0.412	0.470	0.504	0.199	0.301	0.446	0.049	0.088	0.216
ComplEx [37] ◇	0.409	0.469	0.530	0.194	0.297	0.450	0.053	0.120	0.266
RotatE [27]	0.428	0.492	0.571	0.241	0.375	0.533	-	-	-
TuckER [38]	0.443	0.482	0.526	0.226	0.394	0.544	-	-	-
ATTH [55]	0.443	0.499	0.486	0.252	0.384	0.549	-	-	-
				Textual encoding approach					
KG-BERT [28]	0.041	0.302	0.524	-	-	0.420	0.023	0.049	0.241
StAR [45]	0.243	0.491	0.709	0.205	0.322	0.482	-	-	-
GenKGC [33]	0.287	0.403	0.535	0.192	0.355	0.439	0.203	0.280	0.351
GTK	0.449	0.501	0.616	0.291	0.402	0.550	0.210	0.366	0.551

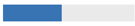
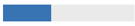
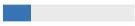
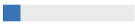
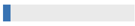
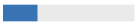
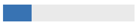
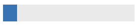
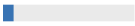
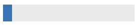
Table 4. Training and inference speed comparison. $|d|$ refers to the length of the entity description. k refers to the number of negative samples of KG-BERT and the beam size for the proposed model. $|\mathcal{E}|$ refers to the number of unique entities in the knowledge graph. We used time under A100 to estimate the training and inference speed on OpenBG500.

For One Triple	**Method**	**Complexity**	**Time under A100**
Training	KG-BERT	$O(\|d\|^2 \times (k+1))$	72 ms
	GTK	$O(\|d\|^2)$	2.01 ms
Inference	KG-BERT	$O(\|d\|^2 \times \|\mathcal{E}\|)$	91,100 s
	GTK	$O(\|d\|^2 \times \|d\|^k)$	0.73 s

5.5. Case Study

To further analyze the proposed model, we randomly sampled some instances and provide case studies. We provide the empirical results with and without knowledge-guided decoding, which indicates that we leveraged the vanilla Beam Search to generate the target entities. As shown in Table 5, we noticed that the proposed model could achieve better generation performance than that of a vanilla Beam Search. For example, the proposed **GTK** model could generate *University of California, Irvine* with the highest probability given the Query (?, student, Michael Chabon). Moreover, we found that the proposed model without knowledge-guided decoding could stop early with the correct—but not very precise—target entity. We think that this was due to the bias from the pre-trained language models, as high-frequency terms may lead to biased predictions. However, the model with knowledge-guided decoding could generate the correct decoding sequence due to prior knowledge.

Table 5. We list a query and the first five entities with their probabilities predicted by **GTK** without entity-aware decoding and the reranking with **GTK**.

Query:`(?,student, Michael Chabon)`		
Rank	**GTK** w/o hierarchical decoding	Probability
1	University of California	
5	University of California, Santa Cruz	
2	**University of California, Irvine**	
3	University of California, San Francisco	
4	University of California, Davis	
Rank	**GTK**	Probability
1	**University of California, Irvine**	
2	University of California, San Francisco	
5	University of Calgary	
4	University of California, Santa Cruz	
3	University of California, Davis	

6. Conclusions and Future Work

In this paper, we proposed a novel model of a generative transformer with knowledge-guided decoding for academic knowledge graph completion. Specifically, we proposed knowledge-aware demonstration and knowledge-guided decoding. We evaluated the proposed approach on four datasets. The extensive experimental results indicate that the proposed model can obtain better performance in academic knowledge graph completion and on popular benchmark datasets, and it can yield a faster inference speed than that of KG-BERT. The success of the proposed approach reveals that treating academic knowledge graph completion as a sequence-to-sequence generation task can be beneficial because knowledge can be inferred directly rather than engaging in the costly scoring of candidate triples. However, the proposed approach still has the limitations of trie tree construction.

In the future, we plan to explore the following: (1) how to obtain the suitable trie tree for candidate knowledge in decoding; (2) how to design specific pre-training objects for knowledge graphs.

Author Contributions: Conceptualization, X.L. and J.B.; methodology, X.L.; software, X.L. and S.M.; validation, X.L. and S.M.; writing, X.L., X.W. and J.B.; review, X.L. and J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 601–610.
2. Zhang, N.; Jia, Q.; Deng, S.; Chen, X.; Ye, H.; Chen, H.; Tou, H.; Huang, G.; Wang, Z.; Hua, N.; et al. Alicg: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, Singapore, 14–18 August 2021; pp. 3895–3905.

3. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
4. Zhang, N.; Chen, M.; Bi, Z.; Liang, X.; Li, L.; Shang, X.; Yin, K.; Tan, C.; Xu, J.; Huang, F.; et al. CBLUE: A Chinese Biomedical Language Understanding Evaluation Benchmark. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 7888–7915.
5. Zhang, N.; Deng, S.; Cheng, X.; Chen, X.; Zhang, Y.; Zhang, W.; Chen, H.; Center, H.I. Drop Redundant, Shrink Irrelevant: Selective Knowledge Injection for Language Pretraining. In Proceedings of the 30th IJCAI, Virtual Event, Montreal, QC, Canada, 19–27 August 2021; pp. 4007–4014.
6. Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; Chen, H. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, 25–29 April 2022; pp. 2778–2788.
7. Ye, H.; Zhang, N.; Deng, S.; Chen, X.; Chen, H.; Xiong, F.; Chen, X.; Chen, H. Ontology-enhanced Prompt-tuning for Few-shot Learning. In Proceedings of the ACM Web Conference 2022, Virtual Event, Lyon, France, 25–29 April 2022; pp. 778–787.
8. Chen, X.; Li, L.; Zhang, N.; Liang, X.; Deng, S.; Tan, C.; Huang, F.; Si, L.; Chen, H. Decoupling Knowledge from Memorization: Retrieval-augmented Prompt Learning. *arXiv* **2022**, arXiv:2205.14704.
9. Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; Chen, H. Reasoning with Language Model Prompting: A Survey. *arXiv* **2022**, arXiv:2212.09597.
10. Deng, S.; Zhang, N.; Zhang, W.; Chen, J.; Pan, J.Z.; Chen, H. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In Proceedings of the Companion Proceedings of The 2019 World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 678–685.
11. Zhang, N.; Bi, Z.; Liang, X.; Cheng, S.; Hong, H.; Deng, S.; Zhang, Q.; Lian, J.; Chen, H. OntoProtein: Protein Pretraining With Gene Ontology Embedding. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
12. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
13. Peroni, S.; Shotton, D. OpenCitations, an infrastructure organization for open scholarship. *Quant. Sci. Stud.* **2020**, *1*, 428–444. [CrossRef]
14. Knoth, P.; Zdrahal, Z. CORE: three access levels to underpin open access. *D-Lib Mag.* **2012**, *18*, 1–13. [CrossRef]
15. Wang, K.; Shen, Z.; Huang, C.; Wu, C.H.; Dong, Y.; Kanakia, A. Microsoft academic graph: When experts are not enough. *Quant. Sci. Stud.* **2020**, *1*, 396–413. [CrossRef]
16. Zhang, Y.; Zhang, F.; Yao, P.; Tang, J. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1002–1011.
17. Jaradeh, M.Y.; Oelen, A.; Farfar, K.E.; Prinz, M.; D'Souza, J.; Kismihók, G.; Stocker, M.; Auer, S. Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge. In Proceedings of the 10th International Conference on Knowledge Capture, Marina Del Rey, CA, USA, 19–21 November 2019; pp. 243–246.
18. Grishman, R. Information extraction. *IEEE Intell. Syst.* **2015**, *30*, 8–15. [CrossRef]
19. Zhang, N.; Ye, H.; Deng, S.; Tan, C.; Chen, M.; Huang, S.; Huang, F.; Chen, H. Contrastive Information Extraction With Generative Transformer. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 3077–3088. [CrossRef]
20. Zhang, N.; Li, L.; Chen, X.; Deng, S.; Bi, Z.; Tan, C.; Huang, F.; Chen, H. Differentiable Prompt Makes Pre-trained Language Models Better Few-shot Learners. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
21. Zhang, N.; Xu, X.; Tao, L.; Yu, H.; Ye, H.; Xie, X.; Chen, X.; Li, Z.; Li, L.; Liang, X.; et al. Deepke: A deep learning based knowledge extraction toolkit for knowledge base population. *arXiv* **2022**, arXiv:2201.03335.
22. Chen, X.; Li, L.; Deng, S.; Tan, C.; Xu, C.; Huang, F.; Si, L.; Chen, H.; Zhang, N. LightNER: A Lightweight Tuning Paradigm for Low-resource NER via Pluggable Prompting. In Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, 12–17 October 2022; Calzolari, N., Huang, C., Kim, H., Pustejovsky, J., Wanner, L., Choi, K., Ryu, P., Chen, H., Donatelli, L., Ji, H., et al., Eds.; International Committee on Computational Linguistics: Praha, Czech Republic, 2022; pp. 2374–2387.
23. Zhang, N.; Chen, X.; Xie, X.; Deng, S.; Tan, C.; Chen, M.; Huang, F.; Si, L.; Chen, H. Document-level Relation Extraction as Semantic Segmentation. In Proceedings of the 30th IJCAI, Virtual Event, Montreal, QC, Canada, 19–27 August 2021.
24. Deng, S.; Zhang, N.; Kang, J.; Zhang, Y.; Zhang, W.; Chen, H. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; pp. 151–159.
25. Lou, D.; Liao, Z.; Deng, S.; Zhang, N.; Chen, H. MLBiNet: A Cross-Sentence Collective Event Detection Network. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Virtual Event, Bangkok, Thailand, 1–6 August 2021; Volume 1: Long Papers, pp. 4829–4839.

26. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. In Proceedings of the NeurIPS, Lake Tahoe, NV, USA, 5–8 December 2013.
27. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the ICLR, New Orleans, LA, USA, 6–9 May 2019.
28. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for Knowledge Graph Completion. *arXiv* **2019**, arXiv:1909.03193.
29. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Virtual Event, 5–10 July 2020; pp. 7871–7880.
30. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
31. Kakade, S.M. A natural policy gradient. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 22.
32. Deng, S.; Chen, H.; Li, Z.; Xiong, F.; Chen, Q.; Chen, M.; Liu, X.; Chen, J.; Pan, J.Z.; Chen, H.; et al. Construction and Applications of Open Business Knowledge Graph. *arXiv* **2022**, arXiv:2209.15214.
33. Xie, X.; Zhang, N.; Li, Z.; Deng, S.; Chen, H.; Xiong, F.; Chen, M.; Chen, H. From Discrimination to Generation: Knowledge Graph Completion with Generative Transformer. In Proceedings of the WWW, Lyon, France, 25–29 April 2022.
34. Zhang, N.; Deng, S.; Sun, Z.; Chen, J.; Zhang, W.; Chen, H. Relation adversarial network for low resource knowledge graph completion. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 1–12.
35. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the AAAI, Austin, TX, USA, 25–30 January 2015.
36. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the ICLR, San Diego, CA, USA, 5–8 December 2015.
37. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. In Proceedings of the ICML, New York, NY, USA, 19–24 June 2016.
38. Balazevic, I.; Allen, C.; Hospedales, T.M. TuckER: Tensor Factorization for Knowledge Graph Completion. In Proceedings of the EMNLP, Hong Kong, China, 3–7 November 2019.
39. Nayyeri, M.; Cil, G.M.; Vahdati, S.; Osborne, F.; Rahman, M.; Angioni, S.; Salatino, A.; Recupero, D.R.; Vassilyeva, N.; Motta, E.; et al. Trans4E: Link prediction on scholarly knowledge graphs. *Neurocomputing* **2021**, *461*, 530–542. [CrossRef]
40. Wang, M.; Wang, S.; Yang, H.; Zhang, Z.; Chen, X.; Qi, G. Is Visual Context Really Helpful for Knowledge Graph? A Representation Learning Perspective. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, China, 20–24 October 2021; pp. 2735–2743.
41. Wu, T.; Khan, A.; Yong, M.; Qi, G.; Wang, M. Efficiently embedding dynamic knowledge graphs. *Knowl.-Based Syst.* **2022**, *250*, 109124. [CrossRef]
42. Zhang, N.; Xie, X.; Chen, X.; Deng, S.; Ye, H.; Chen, H. Knowledge Collaborative Fine-tuning for Low-resource Knowledge Graph Completion. *J. Softw.* **2022**, *33*, 3531. [CrossRef]
43. Wang, Q.; Huang, P.; Wang, H.; Dai, S.; Jiang, W.; Liu, J.; Lyu, Y.; Zhu, Y.; Wu, H. Coke: Contextualized knowledge graph embedding. *arXiv* **2019**, arXiv:1911.02168.
44. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
45. 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 WWW, Virtual Event, Ljubljana, Slovenia, 12–23 April 2021.
46. Zhang, N.; Xie, X.; Chen, X.; Deng, S.; Tan, C.; Huang, F.; Cheng, X.; Chen, H. Reasoning through memorization: Nearest neighbor knowledge graph embeddings. *arXiv* **2022**, arXiv:2201.05575.
47. 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, Dublin, Ireland, 22–27 May 2022.
48. Wang, X.; He, Q.; Liang, J.; Xiao, Y. Language Models as Knowledge Embeddings. *arXiv* **2022**, arXiv:2206.12617.
49. 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 Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 3570–3581.
50. Markowitz, E.; Balasubramanian, K.; Mirtaheri, M.; Annavaram, M.; Galstyan, A.; Ver Steeg, G. StATIK: Structure and Text for Inductive Knowledge Graph Completion. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022, Virtual Event, 10–15 July 2022; pp. 604–615.
51. Shen, J.; Wang, C.; Gong, L.; Song, D. Joint language semantic and structure embedding for knowledge graph completion. *arXiv* **2022**, arXiv:2209.08721.
52. 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, Dublin, Ireland, 22–27 May 2022.
53. Chen, C.; Wang, Y.; Li, B.; Lam, K.Y. Knowledge Is Flat: A Seq2Seq Generative Framework for Various Knowledge Graph Completion. *arXiv* **2022**, arXiv:2209.07299.

54. Liu, R.; Zheng, G.; Gupta, S.; Gaonkar, R.; Gao, C.; Vosoughi, S.; Shokouhi, M.; Awadallah, A.H. Knowledge Infused Decoding. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
55. Chami, I.; Wolf, A.; Juan, D.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the ACL, Virtual Event, 5–10 July 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.