From Discrimination to Generation: Knowledge Graph Completion with Generative Transformer

Xin Xie
Zhejiang University & AZFT Joint Lab for Knowledge Engine Hangzhou, China
xx2020@zju.edu.cn

Ningyu Zhang∗
Zhejiang University & AZFT Joint Lab for Knowledge Engine Hangzhou, China
zhangningyu@zju.edu.cn

Zhoubo Li, Shumin Deng
Zhejiang University & AZFT Joint Lab for Knowledge Engine Hangzhou, China
{zhoubo.li,231sm}@zju.edu.cn

Hui Chen
Alibaba Group Hangzhou, China
weidu.ch@alibaba-inc.com

Feiyu Xiong, Mosha Chen
Alibaba Group Hangzhou, China
{feiyu.xfy,chenmosha.cms}@alibaba-inc.com

Huajun Chen∗
Zhejiang University & AZFT Joint Lab for Knowledge Engine Hangzhou, China
huajunsir@zju.edu.cn

1 INTRODUCTION

Knowledge Graphs (KGs) treat the knowledge in the real world as fact triples in the form of \(\langle\text{subject}, \text{predicate}, \text{object}\rangle\), abridged as \((s, p, o)\), where \(s\) and \(o\) denote entities and \(p\) are the relations between entities, which can benefit a wide range of knowledge-intensive tasks. Knowledge graph completion (KGC) aims to complete the knowledge graph by predicting the missing triples. In this paper, we mainly target link prediction task for KGC based on the powerful pre-trained language models.

Most previous KG completion methods, such as TransE [2], ComplEx [11], and RotatE [9], are knowledge embedding techniques that embed the entities and relations into a vector space and then obtain the predicted triples by leveraging pre-defined scoring functions to those vectors. Recently, some textual encoding methods (e.g., KG-BERT [14]) have been proposed, which utilize the pre-trained language model to encode triples and output the score for each candidate. Obviously, most previous approaches leverage the discrimination paradigm with a pre-defined scoring function to knowledge embeddings. However, such a discrimination strategy has to costly scoring of all possible triples in inference and suffer from the instability of negative sampling. Moreover, those dense knowledge embedding approaches (e.g., TransE) ignore the fine-grained interactions between entities and relations and have to allocate a large memory footprint for the large-scale real-world knowledge graph. Therefore, it is intuitive to find a new technical solution for knowledge graph completion.

In this paper, we take the first step to model the knowledge graph completion with sequence to sequence generation and propose a novel approach GenKGC. To be specific, we represent entities and relations as input sequences and utilize the pre-trained language model to encode triples and output the score for each candidate. Obviously, most previous approaches leverage the discrimination paradigm with a pre-defined scoring function to knowledge embeddings. However, such a discrimination strategy has to costly scoring of all possible triples in inference and suffer from the instability of negative sampling. Moreover, those dense knowledge embedding approaches (e.g., TransE) ignore the fine-grained interactions between entities and relations and have to allocate a large memory footprint for the large-scale real-world knowledge graph. Therefore, it is intuitive to find a new technical solution for knowledge graph completion.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW ’22 Companion, April 25–29, 2022, Virtual Event, Lyon, France © 2022 Association for Computing Machinery. ACM ISBN 978-1-4503-9130-6/22/04...$15.00 https://doi.org/10.1145/3487553.3524238
We convert link prediction to sequence to sequence generation and propose GenKGC, which can reduce the inference time while maintaining the performance.

We propose relation-guided demonstration and entity-aware hierarchical decoding, which can better represent entities and relations and reduce the time complexity of generation.

We report results on two datasets and release a new large-scale KG dataset, OpenBG500, for research purposes.

2 METHOD

2.1 Link Prediction as Seq2Seq Generation

Knowledge graph is defined with entity categories and entity descriptions as a tuple \( G = (E, R, T, C, D) \), where \( E \) represents a set of entities, \( R \) represents relation types, \( T \) refers to a set of triples, \( C \) refers to the entity categories and \( D \) refers to the entity descriptions. For each triple \( t \in T \), there is the form \((e_i, r_j, e_k)\) where \( e_i, e_k \in E \) is the head and tail entity respectively. For each entity \( e_i \in E \), there exists category \( c_i \) to define the \( e_i \) and a text \( d_i \) to describe \( e_i \). To complete missing triples in KGs, link prediction aims at predicting the tail entity given the head entity and the query relation, denoted by \((e_i, r, ?)\).

In this paper, we utilize the standard encoder-decoder architecture for sequence-to-sequence generation. Note that we regard each entity and relation as the sequence of tokens. Concretely, we follow [14] to use plain text to represent the entities and relations instead of unique embedding to bridge the gap between the entities in the knowledge graph and pre-trained language models. To be specific, given a triple with tail entity missing \((e_i, r_j, ?)\), we obtain the description \(d_e, dr_j\) of \(e_i\) and \(r_j\) respectively, and concatenate them together. For example, as shown in Figure 1, "Michael Chabon studied at" (head entity, relation) is the major part of the input sequence, and "UC, Irvine" (target entity) is part of the output sequence. Thus, we have the input sequence of \(<e_i, r_j, ?>\) pair and generate the output sequence of \(e_k\). We leverage BART for model training and inference.

![Architecture of GenKGC](image)

**Figure 1:** Architecture of GenKGC. We augment the input text of entity and relation with demonstrations, and introduce entity-aware hierarchical decoding for fast inference.

### Table 1: Inference and training efficiency comparison

<table>
<thead>
<tr>
<th>For One Triple Method</th>
<th>Complexity</th>
<th>Time under RTX 3090</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>(O(k + 1))</td>
<td>0.08ms</td>
</tr>
<tr>
<td>Training</td>
<td>(O(</td>
<td>d</td>
</tr>
<tr>
<td>GenKGC</td>
<td>(O(</td>
<td>d</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inference</th>
<th>Method</th>
<th>Complexity</th>
<th>Time under RTX 3090</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>(O(</td>
<td>E</td>
<td>))</td>
</tr>
<tr>
<td>GenKGC</td>
<td>(O(</td>
<td>d</td>
<td>^2 \times</td>
</tr>
</tbody>
</table>

2.2 Relation-guided Demonstration

Inspired by prompt-tuning [18], we propose relation-guided demonstration for the encoder. Note that there exist long-tailed distributions in the KGs, for example, there exist only 37 instances of the relation \(film\) in the FB15k-237 dataset. Previous study [18] illustrates that concatenating randomly sampled instances as demonstrations can yield better few-shot performance. Thus, we construct relation-guided demonstration examples \(\{r_{in, train}\}\). To be specific, we sample those demonstrations with the guidance of relation \(r_j\), which consists of several triples with the same relation of input from training set. Formally, we have the final input sequence as:

\[ x = <\text{bos}> \text{demonstration}(r_j) <\text{sep}> d_e, dr_j <\text{sep}> \]

2.3 Entity-aware Hierarchical Decoding

In the vanilla decoding setting, we have to enumerate all entities in the \(E\) and then sort them by the score function. However, this process can be time-consuming, as shown in Table 1 when \(|E|\) is very large, it is costly scoring of all possible triples. In our approach, we follow [3] to use the Beam Search to obtain top-\(k\) entities in the \(E\) (\(k\) is the hyperparameter of beam size). Intuitively, there is no need for negative sampling as we directly optimize by predicting
the correct entity in decoding. To be more specific, given with a triple with tail or head entity missing \((e_i, r_j, ?)\), GenKGC rank each \(e \in E\) by computing a score with an autoregressive formulation:

\[
p_\theta(y \mid x) = \prod_{i=1}^{\lfloor |c| \rfloor} p_\theta(z_i \mid z_{<i}, x) \prod_{i=|c|+1}^{N} p_\theta(y_i \mid y_{<i}, x),
\]

where \(z\) is the set of \(|c|\) tokens in the category \(c\), \(y\) is the set of \(N\) tokens in the textual representation of \(e\).

Since KG contains rich semantic information such as entity types, it is intuitive to constrain the decoding for fast inference. We sample the category with the lowest frequency of occurrence in the training set to constrain the entity decoding since it is challenging to discriminate those low-frequent entities. Then, we add special tokens as types in the vocabulary of pre-trained language model to constrain the decoding. To ensure that the generated entities are inside the entity candidate set, we construct a prefix tree (trie tree) to decode our entity name as shown in Figure 1. Similar to the ordinary sequence-to-sequence model, we optimize GenKGC using a standard seq2seq objective function:

\[
\mathcal{L} = - \log p_\theta(y \mid x)
\]

### 3 EXPERIMENT

Datasets. We evaluate our method on FB15k-237 [10], WN18RR [5], which are widely used in the link prediction literature, and a new real-world large-scale Chinese KG dataset OpenBG500\(^2\). In FB15k-237, descriptions of entities are obtained from the introduction section of the Wikipedia page of each entity. In WN18RR, each entity corresponds to a word sense, and description is the word definition. In OpenBG500, the descriptions of entities and relations come from the e-commerce description page. More details about datasets are listed in Table 3.

\(^2\)OpenBG500 is a subset of open business KG from https://kg.alibaba.com/.

#### Table 3: Summary statistics of benchmark datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Ent</th>
<th># Rel</th>
<th># Train</th>
<th># Dev</th>
<th># Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>WN18RR</td>
<td>40,943</td>
<td>11</td>
<td>86,835</td>
<td>3,034</td>
<td>3,134</td>
</tr>
<tr>
<td>FB15k-237</td>
<td>14,541</td>
<td>237</td>
<td>272,115</td>
<td>17,535</td>
<td>20,466</td>
</tr>
<tr>
<td>OpenBG500</td>
<td>269,658</td>
<td>500</td>
<td>1,242,550</td>
<td>5,000</td>
<td>5,000</td>
</tr>
</tbody>
</table>

Setting. We adopt BART-base as our backbone for comparison with other BERT-based KGC methods like KG-BERT. Following StAR [12], we choose the two kinds of KGE methods as our baseline models, which can be classified as graph embedding approach and textual encoding approach. Grid search is used over the three datasets and the results are reported on the test set with hyperparameters of the best performance determined by the dev set.

Metrics. We evaluate the test set of triples \(T'\) disjoint from the set of training triples \(T\). For inference on a test triple \((e_i, r_j, ?)\), we make sure predict the entity in the candidate set \(E\). We use the metrics of hits@1, hits@3 and hits@10.

Comparison with Discrimination Method. From Table 2, we notice that GenKGC achieves better performance than KG-BERT [14] across all datasets and maintains high speed during inference. The translation-based method like TransE, which treats entities and relations as dense vectors in the same space, will face the memory explosion problem. For the memory cost, TransE has to consume 260M parameters to store the entities and relations for OpenBG500 with more than 260k entities, while pre-trained models (BERT or BART) only utilize 110M parameters, which demonstrates the memory efficiency of our approach. Note that this problem will be more severe when the entities become more numerous because the space complexity is \(O(n)\). For the inference speed, KG-BERT encodes the relational triples with the pre-trained language model and ranks all candidate triples with correct and wrong entities for inference. When the candidate entities set is huge, it is time-consuming for inference; for example, as shown in Table 1, KG-BERT takes about 100100s to the scoring of all possible triples given a single (entity, relation).
Table 4: We list a query and first five entities with their probability predicted by GenKGC w/o entity-aware decoding, and its reranking with GenKGC.

<table>
<thead>
<tr>
<th>Rank</th>
<th>GenKGC w/o hierarchical decoding</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>University of California, Irvine</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>University of California, Davis</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>University of California, San Francisco</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>University of California, Davis</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>University of California, Santa Cruz</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rank</th>
<th>GenKGC</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>University of California, Irvine</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>University of California, Davis</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>University of California, San Francisco</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>University of California, Davis</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>University of California, Santa Cruz</td>
<td></td>
</tr>
</tbody>
</table>

relation) pair. While our method only needs to generate the top-k entities with entity-aware hierarchical decoding, which reduces lots of computing resources.

3.1 Case Study
For different decoding strategies we conduct case study to analyze the result. For GenKGC w/o hierarchical decoding, we utilize the normal beam search to decode the text name of the missing entity. From Table 4, we observe that GenKGC obtain better entity generation results while in normal beam search, the model may stop early at correct but not precise enough answer. We argue that this is caused by the bias of the pre-trained language model (e.g., common token bias) since high-frequent tokens will lead the pre-trained language model to be biased toward certain answers. Our entity-aware hierarchical decoding can constrain the decoding process and mitigate the bias effect caused by pre-trained language models.

4 RELATED WORK
Knowledge Graph Embedding Models. There are lots of methods of KGE are based on knowledge graph embeddings (KGE), which generally leverage an embedding vector in the continuous embedding space to represent the entity and the relation in KG [17]. One kind of KGE methods is translation-based, such as TransE [2], ConE [20], TotatE [9], which consider relations as the mapping function between entities. The other kind of KGE method is semantic matching models, where they get the semantic similarity by using the multi-linear or bilinear product.

Pre-trained Language Models for KGC. Recently, since pre-trained language models, such as BERT [6], have shown significant improvement on several natural language processing tasks, several works use the transformer-based models to tackle the KGC problems [15, 16, 19]. KG-BERT [14] first propose to use BERT for KG by seeing a triple as a sequence and converts KG into a sequence classification problem with the binary cross-entropy object. [7] proposes to use a transformer encoder-decoder model that takes plain text as input and output a structured triple of the information hide in it.

5 CONCLUSION
In this paper, we propose GenKGC, which can reach comparable results while reducing inference and training cost in link prediction with pre-trained models. Experimental results on three benchmark datasets demonstrate the effectiveness of our approach, especially in inference time.

ACKNOWLEDGMENTS
This work is funded by NSFC91846204/NSFCU19B2027, National Key R&D Program of China (Funding No.SQ2018YFC000004), Zhejiang Provincial Natural Science Foundation of China (No. LGG22F030011), Ningbo Natural Science Foundation (2021J190), and Yongjiang Talent Introduction Programme (2021A-156-G).

REFERENCES