Knowledge Is Flat: A Seq2Seq Generative Framework for Various Knowledge Graph Completion

Chen Chen[1,*], **Yufei Wang**[2,*] **Bing Li**[3] and **Kwok-Yan Lam**[1†]

Nanyang Technological University, Singapore[1]
Macquarie University, Sydney, Australia[2]
Centre for Frontier AI Research (CFAR), A*STAR, Singapore[3]
{S190009,kwokyan.lam}@ntu.edu.sg,
yufei.wang@students.mq.edu.au
li_bing@ihpc.a-star.edu.sg

Abstract

Knowledge Graph Completion (KGC) has been recently extended to multiple knowledge graph (KG) structures, initiating new research directions, e.g. static KGC, temporal KGC and few-shot KGC (Ji et al., 2022). Previous works often design KGC models closely coupled with specific graph structures, which inevitably results in two drawbacks: 1) structure-specific KGC models are mutually incompatible; 2) existing KGC methods are not adaptable to emerging KGs. In this paper, we propose `KG-S2S`, a Seq2Seq generative framework that could tackle different verbalizable graph structures by unifying the representation of KG facts into "flat" text, regardless of their original form. To remedy the KG structure information loss from the "flat" text, we further improve the input representations of entities and relations, and the inference algorithm in `KG-S2S`. Experiments on five benchmarks show that `KG-S2S` outperforms many competitive baselines, setting new state-of-the-art performance. Finally, we analyze `KG-S2S`'s ability on the different relations and the *Non-entity Generations* [1].

1 Introduction

Knowledge graph completion (KGC) has been a fundamental task to discover unobserved facts from various knowledge graph (KG) structures, including static KGC (SKGC), temporal KGC (TKGC) and few-shot KGC (FKGC) (Ji et al., 2022). As shown in Figure 1, TKGC (in orange) contains temporal facts with timestamps, while FKGC (in green) predicts the facts with relations that only have limited or zero training instances.

Typically, the solutions for KGC are graph-based, i.e., treating entities and relations as *nodes* and *linkages*. The training and inference of SKGC
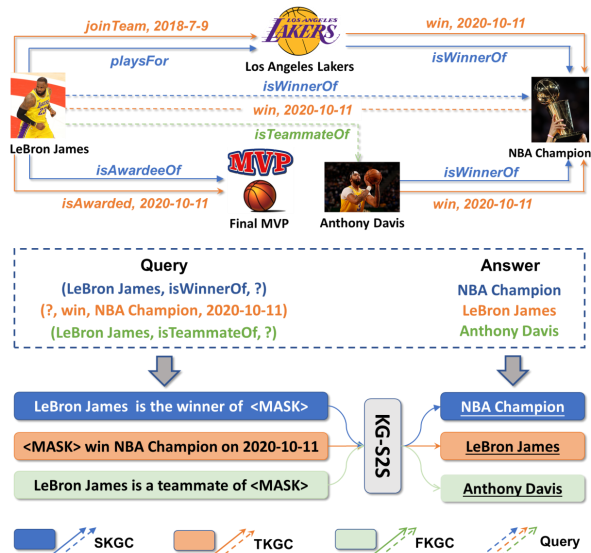

Figure 1: Running examples of Static (SKGC), Temporal (TKGC) and Few-shot (FKGC) Knowledge Graph Completion tasks. Our proposed `KG-S2S` is an unified Seq2Seq framework adaptable to all of these tasks.

models rely on various *transitional relations* over graph paths (Trouillon et al., 2016; Dettmers et al., 2018; Vashishth et al., 2020). TKGC and FKGC methods are further integrated with non-trivial components or learning paradigms to handle the extra temporal information or training requirements. Concretely, TKGC models (Dasgupta et al., 2018; Goel et al., 2020; Lacroix et al., 2020) either construct temporal-specific sub-KG or add additional temporal embeddings into existing SKGC methods. FKGC models apply the additional training scheme (e.g., meta-learning) between the frequent relations and the infrequent ones to the SKGC models (Xiong et al., 2018; Niu et al., 2021). Such a methodological discrepancy leads to a great maintenance cost and being inadaptable to emerging knowledge queries, ingestion, and presents. Naturally, a research question has been raised: *Can we adapt the different forms of KG facts and solve these KGC tasks in a unified framework?*

---

*First two authors contribute equally.
†Corresponding author
[1]Our source code is available at https://github.com/chenchens190009/KG-S2S

Recently, Seq2Seq Pre-trained Language Models (PLM) have shown state-of-the-art performances and high technical homogeneity when dealing with different NLP tasks. Albeit having heterogeneous input and output, the Seq2Seq PLMs covert those tasks into "text-to-text" format, taking the text as inputs and producing another text as outputs (Raffel et al., 2020; Xie et al., 2022a). In addition, PLMs have embedded massive real-world knowledge from the pre-training (Petroni et al., 2019), which is potentially beneficial for the KGC tasks, especially in the data-sparsity scenarios.

Inspired by this, we propose `KG-S2S`, a simple yet effective Seq2Seq PLM framework adaptable to various KG structures. Given a KG query, `KG-S2S` directly generates the target entity text using the common PLM fine-tuning practices. Firstly, to remedy the KG structure information loss caused by the naïve "text-to-text" format, we improve `KG-S2S` via 1) the input representations of entities and relations using *Entity Description*, *Soft Prompt* and *Seq2Seq Dropout*; 2) the constrained inference algorithm empowered by the *Prefix Constraints*; Secondly, we treat all the KG elements (i.e., entity, relation and timestamp) as "flat" text (Figure 1) which enables `KG-S2S` to **i)** handle various *verbalizable* knowledge graph structures; **ii)** generate non-entity text and find novel entities for KGs. We make several improvements on the preliminary attempts of concurrent works (Saxena et al., 2022; Xie et al., 2022b) using Seq2Seq for KGC. Our model adds special treatments to input entity/relation textual representation. This helps to better capture subtle yet key tokens and facilitate the ability to ingest other graph structures.

We conduct experiments on `WN18RR`, `FB15K-237` and `FB15K-237N` for SKGC, `ICEWS14` for TKGC and `NELL-One` for FKGC. `KG-S2S` outperforms several competitive baseline models, including graph-based and PLM-based models, and sets new state-of-the-art performance on all three settings. We conduct ablation studies to show the effectiveness of the proposed components, compare `KG-S2S` with graph-based KGC models at the relation level and finally showcase the *Non-entity Generation* from `KG-S2S` to present its potential in producing novel knowledge triples.

2   Related work

KGC has been studied in the Static, Temporal and Few-shot settings. Previous works often focus on a single setting, while `KG-S2S` fits all three settings without any architecture modifications.

Static KGC   Early SKGC models assign trainable embeddings to each entity and relations (Bordes et al., 2013; Sun et al., 2019). A score function is proposed to evaluate the scores of triples with these embedding. These models learn structural information of a knowledge graph, regardless of the textual information of the entities and relations. Recently, Yao et al. (2019); Wang et al. (2021a); Xie et al. (2022b); Saxena et al. (2022) proposed to encode entity and relation textual knowledge into the model by using PLMs. Instead of calculating scores from embeddings, they train PLMs to produce plausibility scores for KG text representation.

Temporal KGC   Many TKGC models incorporate additional time-specific parameters upon existing KGC methods. Leblay and Chekol (2018), based on Bordes et al. (2013), represents each timestamp with independent embeddings. Dasgupta et al. (2018) resembles Wang et al. (2014), regarding timestamps as hyperplanes for entities to project. Lacroix et al. (2020) considers the score of each triple as canonical decomposition of order 4 tensors in complex domain. Goel et al. (2020) suggests learning dynamic embeddings for entity and relations, transforming part of the embedding with sinusoidal activation of learned frequencies. Han et al. (2021) proposes a systematic framework to improve existing temporal embedding models.

Few-shot KGC   For one-shot learning on relations, Xiong et al. (2018) attempts to seek a matching metric that can be used to discover similar triples given one reference triple. Chen et al. (2019) discovers two kinds of relation-specific meta-information: relation meta, and gradient meta. It uses meta-learning methods to transfer meta-information to low-resource relations. With the help of textual information and PLMs, Wang et al. (2021a) outperforms other few-shot baseline models on the zero-shot relations.

3   Proposed Method

This section first formulates *Knowledge Graph Completion* tasks in Sec. 3.1, then discusses our proposed `KG-S2S` method in Sec. 3.2, 3.3 and 3.4. Figure 2 shows the overview of `KG-S2S`.

Figure 2: The overview of `KG-S2S`. Given the query (LeBron James, is the winner of, ?, ∗), we represent "LeBron James" by joining entity name and description. *Soft Prompt distinguishes* similar relation name and *disentangles* relation-specific information. *Seq2Seq Dropout* randomly masks input words to avoid over-fitting. In the inference, *Prefix Constraints* (Pre. Con.) forces the decoding algorithms to *only* generate valid entity text in the decoder.

3.1 Knowledge Graph Completion

A *Knowledge Graph* (KG) $(\mathcal{E}, \mathcal{R}, \mathcal{T})$ includes an entity set $\mathcal{E}$, a relation set $\mathcal{R}$ and a tuple list $\mathcal{T} = [(h, r, t, m)_1, \cdots, (h, r, t, m)_n]$ where $h, t \in \mathcal{E}$ is head and tail entity, $r \in \mathcal{R}$ is the tuple relation and m is the KG meta-information. *Knowledge Graph Completion* (KGC) predicts the missing entities for the queries $(?, r, t, m)$ or $(h, r, ?, m)$.

The meta-information m denotes different form of contents in different KG settings. As shown in Figure 2, m is represented as null in SKGC, timestamps (e.g., "Jun-19-2014") in TKGC and typing (e.g., Player-Championship) in the KGs providing typing information. Using text representation, KGs with different structures can be converted into an unified format.

3.2 A Seq2Seq Framework for KGC

A Seq2Seq Framework, including an *encoder* and a *decoder*, can be viewed as:

$$P(Y|X) = \prod_{t=1}^{m} P(y_t|X, Y_{<t}) \qquad (1)$$

where X is the input sequence to the Seq2Seq *encoder*, Y is the auto-regressively generated output sequence (i.e., from left to right) and y_0 is the special Begin-of-Sequence Symbol. To apply this Seq2Seq Framework to KGC, given query $(?, r, t, m)$ or $(h, r, ?, m)$ and corresponding ground-truth answer gt, we first encode r, t and m into text. We represent "?" with "<mask>" at the corresponding position to distinguish between $(?, r, t, m)$ and $(h, r, ?, m)$. We then concatenate the text together into X and train `KG-S2S` to generate gt as output sequence Y. The `KG-S2S` training is straightforward: unlike StAR (Wang et al., 2021a) which applied composite objective over the encoder-only PLM, we follow the common practices in fine-tuning Seq2Seq PLMs (i.e.,

Cross-Entropy Loss), directly training `KG-S2S` with positive examples (negative sampling trick is unnecessary to `KG-S2S`). However, this architecture remains two main challenges: *i)* How to *effectively* represent the query in the `KG-S2S` encoder? *ii)* How to *accurately* generate entity text as the answer to the query? Sec. 3.3 and Sec 3.4 answer the above two questions, respectively.

3.3 Entity & Relation Representation

Encoding query $(?, r, t, m)$ and $(h, r, ?, m)$ into "flat" text allows `KG-S2S` to handle various KGs. However, the "flat" text could introduce KG structure loss. To remedy this issue, we further improve `KG-S2S` using the following components.

Entity Description Intuitively, one could represent an entity using its name text (e.g., Anthony Davis) in either compositional or non-compositional form (Li et al., 2018a). However, as `KG-S2S` is initialized from the PLM weights, some specific types of entities (e.g., person, locations) may refer to multiple real-world entities in the large-scale PLM training corpus, introducing noisy ambiguity to `KG-S2S`. To avoid this risk, we additionally introduce entity descriptions to enrich the context information of entities. For example, the textual description about "Lebron James" could be "is an American NBA star". Previous research (Zuo et al., 2018; Lovelace et al., 2021) have shown the utility of the descriptions when integrated with traditional graph-based KGC models. Likewise, we add entity descriptions for both queries and ground-truth answers. At the encoder side, we concatenate entity names and descriptions as the entity representation. At the decoder side, we *train* `KG-S2S` to jointly predict entity names and entity descriptions under the cross-entropy loss. We find using entity descriptions on both sides of `KG-S2S` is beneficial.

KG Soft Prompt In traditional graph-based KGC models (e.g., TransE), KG entities and relations are represented with separated embeddings, while `KG-S2S` represents entities and relations using the shared Seq2Seq PLM parameters. As a consequence, the KG knowledge/patterns regarding similar surface relations or entities (e.g., "film costume design by" and "film production design by") could be mixed together. To tackle this issue and inspired by the recent *Soft Prompt* (Lester et al., 2021), which is a set of trainable embeddings directly fed into the Seq2Seq PLM input, we propose to add additional trainable prompt embeddings for specific entities and relations into X. The separated parameter space could potentially *disentangle* the general KG and element-specific knowledge for `KG-S2S`. However, as entities are equipped with descriptions and *Entity Soft Prompt* introduces a large number of parameters, we only apply the *Soft Prompt* to relations. Specifically, as shown in Figure 2, similar to a recent BERT-based KGC model (Lv et al., 2022), we insert the *Relation Soft Prompt* embeddings $P_{e1}, P_{e2}, P_{r1}, P_{r2} \in \mathbb{R}^{|\mathcal{R}| \times d}$, where d is the `KG-S2S` hidden size, before and after the textual entity and relation name.

Seq2Seq Dropout In our preliminary experiments, we find that `KG-S2S` often learns fast (measured by validation MRR) in the early stage of the model training. We hypothesize that, unlike other NLG tasks where different instances have little textual overlapping, the entity descriptions remain unchanged in different training queries in KGC, which could easily lead to over-fitting. We attempt to increase the original *encoder dropout* for `KG-S2S` training, however, it has little impact on the final performance. Therefore, we impose a more strict *Seq2Seq Dropout* where we randomly select and mask $p\%$ of the input tokens in X when calculating the encoder self-attention module and decoder cross-attention module. Note that the *Relation Soft Prompt* and the "<mask>" token are excluded from this selection process. Compared with the original *encoder dropout*, Seq2Seq dropout takes effect at both encoder and decoder sides. This introduces more diversity to the input query text, thus, better capability of preventing over-fitting.

3.4 KGC Inference

The traditional KGC models g answer a query $(?, r, t, m)$ by first finding the score $g(x, r, t, m)$ $\forall x \in \mathcal{E}$ and then ranking all entities based on

Algorithm 1 Next Candidates (NC): Given Entity Prefix Trie $\mathcal{T}$, Query-GT Prefix Trie Mapping $\mathcal{D}$, query q and the generated tokens Gen; return candidate tokens.

1: **procedure** NC($\mathcal{T}, \mathcal{D}, q, Gen$)
2:   $\mathcal{T}_q \leftarrow \mathcal{D}.get(q)$
3:   $cand \leftarrow \mathcal{T}.next(pre = Gen)$
4:   $rm \leftarrow \mathcal{T}_q.next(pre = Gen)$
5:   $cand \leftarrow cand.remove(rm)$
6:   **return** $cand$

$g(x, r, t, m)$. Naturally, at the inference stage, `KG-S2S` could compute a score for every $x \in \mathcal{E}$. However, this could be computationally expensive because $|\mathcal{E}|$ could be very large (e.g., $|\mathcal{E}|$ is 68,544 in `NELL-One`). Instead, in `KG-S2S`, its encoder takes X as input and then the `KG-S2S` decoder generates the text of entity predictions that are mapped into specific entity ids. These generated entities are further ranked based on their corresponding log cross-entropy loss. We assign -∞ for all entities not generated in the decoding stage.

Decoding Methods Different from general text generation tasks where only one optimal output sequence is required, in KGC, given a query $(?, r, t, m)$ or $(h, r, ?, m)$, there could be multiple valid entities. To generate K valid entity candidates, we deploy the standard beam search algorithm with beam width K because it naturally produces different entity text in each beam with high likelihood. In contrast, random sampling often provides low-quality answers due to its randomness in decoding and the outputs of diverse beam search are distorted due to its diversity encouragement term.

Prefix Constraints The flexible auto-regressive generation may produce entities that do not exist in $\mathcal{E}$, which could reduce the number of valid entity candidates in the decoding. To avoid this scenario, we propose *Prefix Constraints* to control the `KG-S2S` decoder to generate valid tokens given prefix sequences p. For example, given $\mathcal{E} = $ {"Grammy Award for Best Rock Song", "Grammy Award for Best Music Video"} and $p = $ [Grammy, Award, for, Best], the *Prefix Constraints* only allow "Rock" and "Music" to be generated in the next step. To enable effective decoding, we propose to use Trie (Cormen et al., 2009) to extract appropriate next tokens. As suggested in Algorithm 1, given the generated prefix, we first extract all possible

tokens using the Entity Prefix Trie $\mathcal{T}$ and then remove the entities that are the ground-truth to the query q in the training data using the Query-GT Prefix Trie $\mathcal{T}_q$.

4 Experiment

In this section, we evaluate **KG-S2S** against competitive baselines in the following KGC datasets: **WN18RR** (Dettmers et al., 2018) (SKGC), **FB15K-237** (Toutanova and Chen, 2015) (SKGC) **FB15K-237N** (Lv et al., 2022) (SKGC) and **ICEWS14** (García-Durán et al., 2018) (TKGC) and **NELL-One** (Xiong et al., 2018) (FKGC).

4.1 Experimental Settings

Dataset **WN18RR** and **FB15K-237** are improved version of WN18 and FB15k (Bordes et al., 2013) respectively, where all inverse relations are removed to avoid data leakage. **FB15K-237N** further removes **FB15K-237**'s concatenated relations caused by Freebase mediator nodes (Akrami et al., 2020) to avoid Cartesian production relation issue. **ICEWS14** refers to 2014 political facts from the Integrated Crisis Early Warning System database (Boschee et al., 2015). **NELL-One** is a few-shot KGC dataset derived from NELL (Carlson et al., 2010). Following Wang et al. (2021a), we reformat **NELL-One** so that the dev/test relations never appear in the train set. More details can be found in Appendix A.

Implementation details We initialize **KG-S2S** using the T5-base model (Raffel et al., 2020), and optimize **KG-S2S** with Adam (Kingma and Ba, 2015). We use T5 default settings in our experiments for all benchmarks and follow the *filtered setting* proposed in Bordes et al. (2013) to evaluate our model. More implementation details and optimal hyperparameters can be found in Appendix B.

Evaluation Protocol We remove the duplicated entities from the output. In the non-constrained decoding method, we further remove non-entity generations. The performance of our model is reported on the standard KGC metrics: Mean Reciprocal Rank (MRR), and Hits@1,3,10 (Hits@1,5,10 in **NELL-One** to follow previous works). For each test triple (h, r, t, m), we rank all entities for the query $(h, r, ?, m)$ and $(?, r, t, m)$. We then aggregate the ranking for ground-truth entity and report the mean reciprocal rank (MRR) and the proportion of ground-truth entities ranked in the top n (H@n).

To handle the equal score scenarios, we use the RANDOM mode proposed in Sun et al. (2020) to determine the rank of entities. Model is selected by MRR value on valid set.

4.2 Experimental results

Static KGC We compare our results with various graph-based and PLM-based methods on the SKGC settings. Experimental results are summarized in Table 1. On **WN18RR** and **FB15K-237**, **KG-S2S** achieves state-of-the-art or competitive performance. In the comparison of PLM-based methods, **KG-S2S** outperforms previous work by a substantial margin. Specifically, we see 13% (from 0.508 to 0.574) relative MRR improvement on **WN18RR**, and 16% (from 0.296 to 0.336) on **FB15K-237**. Compared with graph-based methods, **KG-S2S** consistently obtains performance gain on **WN18RR**, though maintaining a modest result on **FB15K-237**.

According to Akrami et al. (2020); Lv et al. (2022), **FB15K-237** contains many oversimplified unrealistic cartesian product relations (CPR), which improperly improves the model accuracy. For instance, the multiary fact "average low temperature in Tokyo is 34 degrees Fahrenheit in January" has been decomposed into multiple CPR facts (Tokyo, climate./month, January) and (Tokyo, climate./average_min_temp, 34), which are obviously unrealistic and semantically meaningless. We note that RotatE achieves higher overall performance than **KG-S2S** on **FB15K-237**.

However, after breaking down the performance on CPRs and non-CPRs in Table 2, we surprisingly find that our proposed **KG-S2S** has distinct advantages on non-CPR (MRR 0.363 vs. 0.338). That is, leading performance of RotatE is due to the facts with CPR, while **KG-S2S** has demonstrated its advantages in realistic relations (i.e., non-CPRs). Methodologically, RotatE is a typical graph-based model, while KG-S2S regards KGs as plain text with structure-aware components. Graph-based models are good at predicting simple structure yet inferior in absorbing KGs text. This explains why RotatE performs better on FB15k-237 dataset which is rich in cartesian product relations (CPRs, simple synthesized yet less textually meaningful relations), while worse on non-CPR datasets like FB15K-237N. This further motivates us to compare **KG-S2S** with other KGC methods on **FB15K-237N** which only has facts with non-CPR.

	WN18RR				FB15K-237				FB15K-237N			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Graph-Based Methods												
TransE (Bordes et al., 2013)	.243	.043	.441	.532	.279	.198	.376	.441	.255	.152	.301	.459
DistMult (Yang et al., 2015)	.444	.412	.470	.504	.281	.199	.301	.446	.209	.143	.234	.330
ComplEx (Trouillon et al., 2016)	.449	.409	.469	.530	.278	.194	.297	.450	.249	.180	.276	.380
ConvE (Dettmers et al., 2018)	.456	.419	.470	.531	.312	.225	.341	.497	.273	.192	.305	.429
RotatE (Sun et al., 2019)	.476	.428	.492	.571	.338	.241	.375	.533	.279	.177	.320	.481
CompGCN (Vashishth et al., 2020)	.479	.443	.494	.546	.355	.264	.390	.535	.316	.231	.349	.480
PLM-Based Methods												
KG-BERT (Yao et al., 2019)	.216	.041	.302	.524	-	-	-	.420	.203	.139	.201	.403
MTL-KGC (Kim et al., 2020)	.331	.203	.383	.597	.267	.172	.298	.458	.241	.160	.284	.430
StAR (Wang et al., 2021a)	.401	.243	_.491_	**.709**	_.296_	.205	.322	_.482_	-	-	-	-
PKGC (Lv et al., 2022)	-	-	-	-	-	-	-	-	_.307_	_.232_	_.328_	_.471_
GenKGC (Xie et al., 2022b)	-	.287	.403	.535	-	.192	_.355_	.439	-	-	-	-
KGT5 (Saxena et al., 2022)	_.508_	_.487_	-	.544	.276	_.210_	-	.414	-	-	-	-
KG-S2S (Ours)	**.574**	**.531**	**.595**	_.661_	**.336**	**.257**	**.373**	**.498**	**.353**	**.282**	**.385**	**.495**

Table 1: Results of static KGC. **WN18RR** and **FB15K-237** results are taken from Wang et al. (2021a). **FB15K-237N** results are taken from (Lv et al., 2022). The uncovered results of graph-based methods are obtained through hyperparameter tuning with LibKGE (Broscheit et al., 2020) and PLM-based methods through official implementations. The best PLM-based method results are in bold and the second best results are in underline.

	relations	MRR	H@1	H@3	H@10
RotatE	CPR	.337	.232	.374	.552
	non-CPR	.340	.254	.376	.504
	all	.338	.241	.375	.533
KG-S2S	CPR	.318	.234	.355	.493
	non-CPR	.363	.292	.398	.504
	all	.336	.257	.373	.498

Table 2: Evaluation of cartesian product relations (CPRs) and non-cartesian product relations (non-CPRs) on **FB15K-237**

	MRR	H@1	H@3	H@10
Graph-Based Methods				
TTransE (Leblay and Chekol, 2018)	.255	.074	-	.601
HyTE (Dasgupta et al., 2018)	.297	.108	.416	.655
ATiSE (Xu et al., 2019)	.550	.436	_.629_	_.750_
DE-SimplE (Goel et al., 2020)	.526	.418	.592	.725
Tero (Xu et al., 2020)	_.562_	.468	.621	.732
TComplEx (Lacroix et al., 2020)	.560	_.470_	.610	.730
TNTComplEx (Lacroix et al., 2020)	.560	.460	.610	.740
T+TransE (Han et al., 2021)	.553	.437	.627	**.765**
T+SimplE (Han et al., 2021)	.539	.439	.594	.730
PLM-Based Methods				
KG-S2S (Ours)	**.595**	**.516**	**.642**	.737

Table 3: Results of temporal KGC on **ICEWS14**. All the results are from original papers.

As shown in Table 1, **KG-S2S** obtains the best results compared with graph-based and PLM-based baselines at all metrics. In particular, **KG-S2S** achieves an absolute Hit@1 increase of 5.0% over second best method Wang et al. (2021a).

The overall SKGC results confirms that, by taking advantage of entity and relation textual representation, **KG-S2S** is capable of capturing more accurate semantics of KG facts, and employ them for inference.

Temporal KGC To evaluate **KG-S2S**'s ability of handling additional meta-information in KG, we conduct the experiment on the TKGC benchmark **ICEWS14**. The results are shown in Table 3. Our proposed **KG-S2S** obtains a new state-of-the-art result on MRR and Hit@1,3 while achieving comparative performance on Hit@10. This result confirms that **KG-S2S** can learn additional temporal meta-information from pure textual form. We observe that our result on Hit@10 is lower than several existing methods. This could be explained by the low quality of entities in **ICEWS14**, which

only includes the "sector" and "country" of the entities. These entity descriptions are much less informative than the ones in the SKGC benchmark. We believe that the performance of **KG-S2S** could be further improved when more informative entity descriptions are available.

Few-shot KGC Finally, we verify **KG-S2S**'s ability in few-shot learning in the **NELL-One** benchmark, as shown in Table 4. Following Wang et al. (2021a), we conduct the evaluation under zero-shot setting (i.e, evaluation relations never appear in the training set). Surprisingly, **KG-S2S** is able to achieve superior performance than all the variations of previous graph-based models, which transfer knowledge from the training data to the evaluation relations (i.e., one-shot and five-shot meta learning). In addition, compared with the PLM-based StAR model, **KG-S2S** also obtains higher performance with considerable margins in

	N-Shot	MRR	H@1	H@5	H@10
Graph-Based Methods					
GMatching$_{ComplEx}$♣	Five	.20	.14	.26	.31
MetaR$^♡$	Five	.26	.17	.35	.44
GMatching$_{TransE}$♣	One	.17	.12	.21	.26
GMatching$_{DistMult}$♣	One	.17	.11	.22	.30
GMatching$_{ComplEx}$♣	One	.19	.12	.26	.31
MetaR$^♡$	One	.25	.17	.34	.40
MTransH$^△$	One	**.31**	.21	**.41**	.48
PLM-Based Methods					
StAR♠	Zero	.26	.17	.35	.45
KG-S2S (Ours)	Zero	**.31**	**.22**	**.41**	**.49**

Table 4: Results of few-shot KGC on NELL-One. △ (Niu et al., 2021). ♣ (Xiong et al., 2018), ♡ (Chen et al., 2019) and ♠ (Wang et al., 2021a).

terms of all the metrics. In particular, Hit@1 performance is boosted from 0.17 to 0.22, around 29% relative improvement. This remarkable performance gain could own to the following aspects: 1) the prior knowledge contained in PLM; 2) **KG-S2S**'s capability to transfer the knowledge from the training relations to the unseen ones.

	PW	Description		Soft Prompt		S2S. Drop	MRR	H@10
		SRC	TGT	REL	ENT			
Baseline	✓	-	-	-	-	-	.280	.416
	✓	✓	-	-	-	-	.326	.453
	✓	✓	✓	-	-	-	.350	.478
	✓	✓	✓	✓	-	-	.350	.486
	✓	✓	✓	-	✓	-	.338	.468
KGT5	-	✓	✓	-	-	-	.226	.335
	-	✓	✓	✓	-	✓	.233	.341
KG-S2S	✓	✓	✓	✓	-	✓	**.353**	**.495**

Table 5: Ablation for the **KG-S2S** Input Components on **FB15K-237N**. PW denotes pretrained weight. SRC and TGT denote source and target description. REL and ENT denote relation-specific and entity-specific soft prompts. S2S.Drop denotes Seq2Seq dropout.

4.3 Ablation Study

In this section, we conduct ablation studies to show the contributions of each of our proposed components. Table 5 shows the impact of input components and Figure 3 shows the ablation study of decoding components in **KG-S2S**.

Source and target description The descriptions of entities enrich their context information and resolve the ambiguity issue. As shown in Table 5, adding source and target description can separately improve MRR (i.e., from 0.280 to 0.350) and Hit@10 (i.e., from 0.416 to 0.478). This suggests that to achieve optimal performance, it is important to inject entity descriptions at **KG-S2S**'s encoder

and decoder, *simultaneously*.

Soft Prompt Soft prompt allows **KG-S2S** to recognize entities and relations as atomic concepts. Adding relation Soft Prompt successfully boosts Hit@10 from 0.478 to 0.486. However, the Entity Soft Prompt has a negative effect for **KG-S2S**, degrading MRR and Hit@10 by 0.12 (from 0.350 to 0.338) and 0.1 (0.478 to 0.468), respectively. We argue this phenomenon occurs for at least two reasons: 1) The entity descriptions have already enriched the entity context information and consequently made entities distinguishable; 2) Entity Soft Prompt introduces massive amounts of embeddings, which may weaken **KG-S2S**'s ability to learn from natural language.

Seq2Seq Dropout Seq2Seq dropout applies a random masking mechanism on the encoder input mask. It is observed that Seq2Seq dropout is able to deliver consistent improvement on both MRR and Hit@10. This result practically justifies the effectiveness of this implementation. We believe the advance is derived from the diversified input data generated by Seq2Seq dropout, which helps **KG-S2S** to avoid potential over-fitting risk.

Campared with KGT5 KGT5 is trained on a random initialized Seq2Seq structure to fully adapt KG training data. However, learnt from large pretraining corpus, pretrained weights contains rich linguistic knowledge and simply dropping them may weaken the model's ability of ingesting nature language. The large performance gap between KGT5 and **KG-S2S** indicates pretrained weights is critical for KGC models with Seq2Seq backbones .
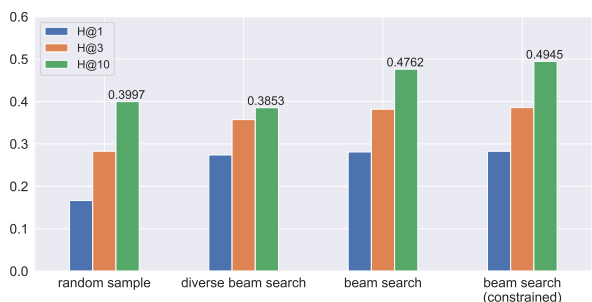

Figure 3: Comparison for decoding methods on **FB15K-237N**.

Impact of Decoding Methods In Figure 3, we investigate the impact of decoding methods, including random sampling, diverse beam search, beam search and *Prefix Constraints* beam search (de-

scribed in Sec. 3.4) in `KG-S2S`. The performances of random sampling and diverse beam search are much worse than the standard beam search algorithm. This is mostly because random selection in sampling and diversity encouragement terms in diverse beam search negatively affect the quality of generated entity text. Whilst standard beam search always keeps and derives the candidates with the highest beam score from `KG-S2S`. We find that applying our *Prefix Constraints* to the beam search algorithm further improves the `KG-S2S` performance (i.e., 0.02 Hit@10 improvement). *Prefix Constraints* control `KG-S2S` to only generate valid entity text with little computation overhead.

4.4 Discussion

Comparison with previous SOTA PLM-based methods From Table 1 and Table 4, `KG-S2S` outperforms previous SOTA encoder-only StAR methods on MRR and Hit@1,3. We argue two advantages contribute to this result: 1) Pretraining / finetuning consistency. StAR employs composite training objectives at the entity level, while its backbones (i.e. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) are trained with token-level cross-entropy loss. This mismatch may weaken the representation ability of PLM. In contrast, `KG-S2S` follows the common PLM finetuning practices, allowing better knowledge transfer from PLM. 2) Information interaction. StAR uses a two-branch Siamese architecture (Chopra et al., 2005) to encode the query text and answer text as two separated vectors, and calculates their dot-production as the score for ranking. Instead of compressing the them separately, `KG-S2S` interacts query and answer in the cross-attention module of `KG-S2S` decoder, auto-regressively. With more textual exposure and interaction, `KG-S2S` decoder trends to predict more accurate entities.

Relations Analysis Figure 4 shows the top-3 and bottom-3 relations regarding the MRR difference between `KG-S2S` and RotatE. The top-3 relations are *(sports team) location*, *(location) contains* and *(Netflix) genre*, which refer to the real-world knowledge. The possible reason is such knowledge has already been obtained from the pre-training corpus by the Seq2Seq PLM. In contrast, the bottom-3 relations are *(film) production company*, *(film) executive produced by* and *(film) produced by*, which, surprisingly, are all relevant to the film industry. This could be because these relations are all linked
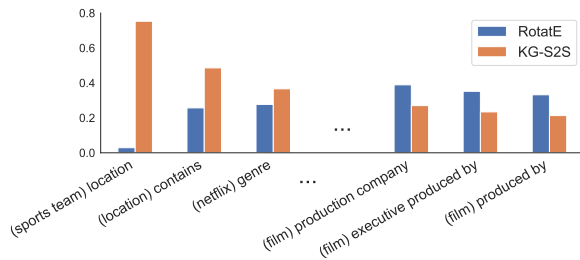

Figure 4: The top-3 (left) / bottom-3 (right) relations regarding the MRR difference between `KG-S2S` and RotatE on the `FB15K-237N` benchmark. To maintain stable results, we select the relations with at least 0.5% facts, and sort them by the MRR difference.

to the person and company names that may have multiple references (i.e., different people and companies could share the same names) in the PLM pre-training corpus. In addition, we find that some of the relations are semantically overlapping. For example, the `FB15K-237N` includes both relation *(film) executive produced by* and *(film) produced by*. After being trained with the fact *(Hulk, (film) executive produced by, Stan Lee)*, `KG-S2S` generates *Stan Lee* as the top-1 candidate for the query *(Hulk, (film) produced by, ?)*. However, the ground-truth entity set doesn't include *Stan Lee*. This scenario has no effect on the traditional graph-based KGC models because they do not access the text at all. Similar cases also occur between *(film) written by* and *(film) story by*, *(people) profession* and *(people) specialization of*. This issue is caused by the fact that previous KGC benchmarks i) are not fully verified by experts; ii) are based on the closed-world assumption (CWA) (Keet, 2013). We leave KGC benchmarks improvement as future work.

Queries	Prediction	GT
(RoboCop, (film) genre, ?)	Superhero film	Thriller
(Amber Riley, profession, ?)	Vocalist	Actor-GB
(? (location) contains, Israel)	Greater Middle East	Eurasia
(?, ethnicity, M. Night Shyamalan)	Malayalam people	Indian American

Table 6: Case study for Non-entity generations. GT stands for ground-truth answer.

Non-entity Generation Without *Prefix Constraints* module, `KG-S2S` can generate non-entity text. As shown in Table 6, some of the non-entity generations are also meaningful answers to the query. In the first example, Superhero film and Thriller are both semantically correct answer. In the second one, Amber Riley is actually considered as an actor and a vocalist by the public. In addition, the third and fourth examples show that `KG-S2S`

can derive more fine-grained answers. For example, *Israel* is specifically located in the *Greater Middle East* and *M. Night Shyamalan* is an *Indian American*, born in a Malayalam-speaking Indian city. These newly generated entities could be potentially applied to improve the KGC model performance via a data augmentation procedure (Wang et al., 2022). The expert knowledge to determine the plausibility of non-entity generations is given by the corresponding entries from Wikipedia, e.g. the Wikipedia profile for Amber Riley [2].

The Effect of Beam Width Beam width determines the number of generations for each query, thus it has potentially significant impact on the `KG-S2S` performance. In Figure 5, we study how beam width affects the final performance by evaluating `KG-S2S` under different beam width. In general, `KG-S2S` achieves higher MRR as the beam width increases, whilst the performance gain becomes flat after 40 beams (red bar). As inference time goes linearly with beam width, we choose beam size 40 in `KG-S2S` to trade-off between model performance and inference cost.
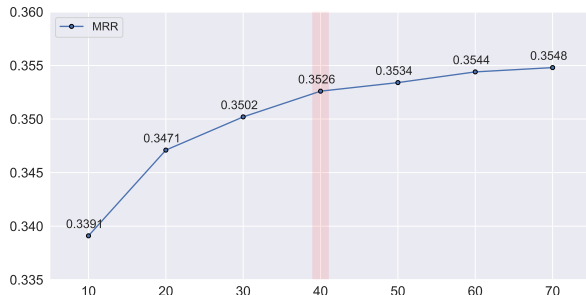

Figure 5: The effect of beam width to KS-S2S

Parameter Size Table 7 compares the model performance and parameter size between `KG-S2S` and StAR. Compared with StAR (354M trainable parameters), `KG-S2S` is based on a smaller T5-base backbone (220M trainable parameters, 1.6x less), while it achieves better performance with a relatively large margin (MRR 0.274 vs. 0.353). We further run `KG-S2S` using T5-small backbone (60M trainable parameters, 5.9x less). This variant of `KG-S2S` obtain slightly lower result (0.351 on MRR), but still outperforms StAR with substantial margin. The results suggest **i)** `KG-S2S` is not sensitive to the size of PLM; **ii)** `KG-S2S` is parameter-efficient.

Model	Size	MRR	H@10
StAR	354M	.274	.455
`KG-S2S` (small)	60M	.351	.485
`KG-S2S` (base)	220M	.353	.495

Table 7: Comparison of model performance and parameter size between `KG-S2S` and StAR on `FB15K-237N`.

Parameter Growth Since `KG-S2S` represents KG elements (e.g. entities, relations and timestamps) as simple textual sequences, all KG tuples share the same vocabulary and language model parameters. As the training KG grows, the parameters of `KG-S2S` only increase due to the relations *Soft Prompt* with the complexity of $O(|\mathcal{R}| \cdot d)$ where d is the hidden size of Seq2Seq Pre-trained language model. On the contrary, traditional graph-based models represent entities, relations and other meta-information with distinct embeddings and the parameter growth of these models is $O((|\mathcal{E}| + |\mathcal{R}|)d)$. As $|\mathcal{R}| \ll |\mathcal{E}|$, the growth could be negligible and the parameter size of `KG-S2S` remains nearly constant given KGs with any size.

5 Conclusion and Future Work

In this paper, we present `KG-S2S` for various knowledge graph completion tasks. By converting different kinds of KG structures into "text-to-text" format, `KG-S2S` can directly produce the text of target predicted entities. Experimental results demonstrate that `KG-S2S` outperforms competitive baseline models in various KGC settings. In the future, we would explore extending `KG-S2S` to other Seq2Seq PLMs, such as BART (Lewis et al., 2020) and MASS (Song et al., 2019). In addition, it is interesting to combine `KG-S2S` with other knowledge-intensive NLP tasks, such as conversation recommendation (Li et al., 2018b) and commonsense generation (Wang et al., 2021b) in the Seq2Seq framework, and see if the KG knowledge could benefit these downstream tasks.

Acknowledgement

References

Farahnaz Akrami, Mohammed Samiul Saeef, Qingheng Zhang, Wei Hu, and Chengkai Li. 2020. Realistic re-evaluation of knowledge graph completion methods: An experimental study. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 1995–2010. ACM.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data. *Harvard Dataverse*.

Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. 2020. Libkge - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 165–174. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1306–1313. AAAI Press.

Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 4216–4225. Association for Computational Linguistics.

S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2001–2011. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. Association for Computational Linguistics.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3988–3995. AAAI Press.

Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021. Time-dependent entity embedding is not all you need: A re-evaluation of temporal knowledge graph completion models under a unified framework. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8104–8118. Association for Computational Linguistics.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514.

C. Maria Keet. 2013. *Closed World Assumption*, pages 415–415. Springer New York, New York, NY.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language

models. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1737–1743. International Committee on Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, pages 1771–1776. ACM.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Bing Li, Xiaochun Yang, Bin Wang, Wei Wang, Wei Cui, and Xianchao Zhang. 2018a. An adaptive hierarchical compositional model for phrase embedding. In *IJCAI*, pages 4144–4151.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018b. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Justin Lovelace, Denis Newman-Griffis, Shikhar Vashishth, Jill Fain Lehman, and Carolyn Rosé. 2021. Robust knowledge graph completion with stacked convolutions and a student re-ranking network. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1016–1029, Online. Association for Computational Linguistics.

Xin Lv, Yankai Lin, Yixin Cao, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022. Do pretrained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3570–3581, Dublin, Ireland. Association for Computational Linguistics.

Guanglin Niu, Yang Li, Chengguang Tang, Ruiying Geng, Jian Dai, Qiao Liu, Hao Wang, Jian Sun, Fei Huang, and Luo Si. 2021. Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 213–222. ACM.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. *CoRR*, abs/2203.10321.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha P. Talukdar, and Yiming Yang. 2020. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5516–5522. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. Structure-augmented text representation learning for efficient knowledge graph completion. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 1737–1748. ACM / IW3C2.

Yufei Wang, Ian Wood, Stephen Wan, Mark Dras, and Mark Johnson. 2021b. Mention flags (MF): Constraining transformer-based text generators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 103–113, Online. Association for Computational Linguistics.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,*

July 27 -31, 2014, Québec City, Québec, Canada, pages 1112–1119. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pier-ric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir R. Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022a. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *CoRR*, abs/2201.05966.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022b. From discrimination to generation: Knowledge graph completion with generative transformer. *CoRR*, abs/2202.02113.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1980–1990. Association for Computational Linguistics.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Jens Lehmann, and Hamed Shariat Yazdi. 2019. Temporal knowledge graph embedding model based on additive time series decomposition. *CoRR*, abs/1911.07893.

Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. Tero: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1583–1593. International Committee on Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Yukun Zuo, Quan Fang, Shengsheng Qian, Xiaorui Zhang, and Changsheng Xu. 2018. Representation learning of knowledge graphs with entity attributes and multimedia descriptions. In *Fourth IEEE International Conference on Multimedia Big Data, BigMM 2018, Xi'an, China, September 13-16, 2018*, pages 1–5. IEEE.

A Dataset

ICEWS14 This dataset doesn't include any entity descriptions. As a result, we find the original data source[3] and create the description by combining 'sector' and 'country' entries for each entity.

NELL-One To conduct zero-shot learning for this dataset, we follow Wang et al. (2021a) to reformat the raw dataset so that the relations in the dev/test sets do not appear in the train set. Additionally, we observe that textual representations of entities and relations are written in lower letters. To avoid pretrain-finetune data format mismatch, we further capitalize the surface words for each entity name. Dataset statistics are shown in Table 8.

Dataset	Setting	$|\mathcal{E}|$	$|\mathcal{R}|$	|Train|	|Valid|	|Test|
WN18RR	SKGC	40,943	11	86,835	3,034	3,134
FB15K-237	SKGC	14,541	237	272,115	17,535	20,466
FB15K-237N	SKGC	14,541	93	87,282	7,041	8,226
ICEWS14	TKGC	6,869	230	72,826	8,941	8,963
NELL-One	FKGC	68,544	358	189,635	1,004	2,158

Table 8: Statistics of the Datasets.

All of these datasets are open-source English-written sources without any offensive content. They are introduced only for research use.

B Inplementation details

We implement our **KG-S2S** using PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020), and assess it on a single GPU (Tesla V100).

Model Input and Output We follow the T5 standard unsupervised training paradigm. We form the query texts by masking the target entities with T5 default special tokens. Answer texts are also wrapped by the T5 special tokens. We use square brackets around descriptions to distinguish them from entity names. A special separation token "|" is inserted to separate entity, relation and meta-information. During the inference stage, our model generates the raw text, and we remove the wrapping

special tokens and corresponding entity descriptions with regular expression, remaining the entity names as model predictions. Practical results suggest that the predicted entities can be determined by the entity names, so it is unnecessary to generate all the descriptions. Consequently, we perform an early stopping generation strategy, that is, the generation process will be stopped if the model outputs reach maximum entity name length.

Seq2Seq Dropout Seq2Seq dropout is applied on the encoder input mask, randomly flipping the values from 1 to 0. Note that Seq2Seq dropout excludes the positions carrying special meanings, i.e. separation tokens, mask tokens and soft prompt.

Hyperparameters In terms of hyperparameters, we select the batch size from {32, 64, 128}, learning rate from {5e-3, 1e-3, 5e-4}, description length from {10, 40, 80}, Seq2Seq dropout from {0.0, 0.1, 0.2, 0.3}. The optimal configurations are displayed in Table 9

	batch size	learning rate	SRC/TGT desc.	S2S.Drop.
WN18RR	64	1e-3	40/40	0.1
FB15K-237	32	1e-3	80/80	0.2
FB15K-237N	32	1e-3	80/80	0.2
ICEWS14	32	5e-4	40/40	0.1
NELL-One	128	5e-4	0/0	0.0

Table 9: Hyperparameters for **KG-S2S**. SRC/TGT desc. denotes source and target description length. S2S.Drop denotes Seq2Seq dropout.

---

[3] https://dataverse.harvard.edu/dataverse/icews