Reducing Bidirectional Link Prediction Gaps with Dual Transformer Encoders for Knowledge Graph Completion

Anonymous ACL submission

Abstract

Knowledge graph is a set of triplets, *i.e.,* (head, relation, tail), that plays a crucial role in machine intelligence, but generally suffers from incompleteness. The knowledge graph completion task aims to predict the missing entity given the other two instances in an incomplete triplet. Previous approaches *e.g.,* StAR, exploit the associated textual context of triplets to improve prediction accuracy by using pre-trained language models. Despite achieving performance improvements, they are inclined to aggravate the performance gap between bidirectional predictions due to the unbalanced attention paid to the head or tail entity. In this paper, we propose a dual Transformer encoding framework combined with a semantics alignment mechanism to balance the roles played by head and tail entities, such that alleviating the performance gaps. To further improve performance, a hard negative sampling strategy is further introduced to train the model, with a theoretical analysis provided to prove its effectiveness. Extensive experiments show that our model surpasses the current state-of-the-art models on three public datasets while successfully decreasing performance gaps between bidirectional predictions.

1 Introduction

Knowledge graph (KG), as a large-scale knowledge database, is often represented as a multi-relational graph, in which entities and relations are denoted as nodes and edges, respectively. KGs are ubiquitous in many information systems, with applications ranging from question answering (Huang et al., 2019), search engines (Xiong et al., 2017) to recommendation systems (Gao et al., 2020) etc. However, as illustrated in Figure 1, KGs in practical applications are mostly incomplete, that is, large amount of links between entities are missing. Therefore, completing KGs by predicting the missing links or inferring the missing entities is of great practical importance.

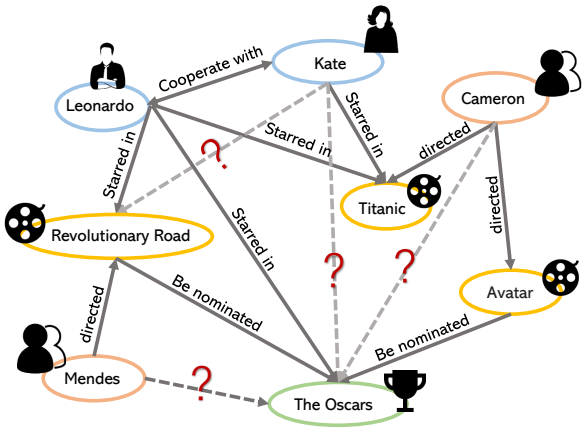

Figure 1: An example of a movie knowledge graph, where the solid line represents a clear relationship, and the dashed line represents a missing relationship.

To complete a knowledge graph, a widely adopted approach is to leverage the connection structure among entities and relations in the graph. Typical examples of models along this line include TransE (Bordes et al., 2013), ComplEx (Trouillon et al., 2016), RotatE (Sun et al., 2019), ConvE (Dettmers et al., 2018), SACN (Shang et al., 2019), KBAT (Nathani et al., 2019) and AttH (Chami et al., 2020) etc. These methods make use of structural information in KGs to predict the missing head in the case of (?, relation, tail) or tail in (head, relation, ?). Although significant performance improvements have been observed, the performance of these methods are known to be significantly limited by low connectivity of sparse KGs.

Considering the entities and relations in KGs are often described by words with clear semantics, it is natural to leverage the associated texts in KG to help the completion. In KG-BERT (Yao et al., 2019), the words from entities and relation in a triplet are concatenated into a sequence and then fed into the BERT(Devlin et al., 2019) encoder to produce a representation, based on which a probability of faithfulness for the triplet is computed. To evaluate the faithfulness of a triplet, KG-BERT re-

quires to pass it through the entire BERT encoder, which is computationally expensive. To reduce the complexity, StAR (Wang et al., 2021) recently proposed to use an asymmetric Siamese-structured encoder to avoid the passing at every evaluation. It is achieved by splitting every triplet into two asymmetric segments, with the first segment composed of words from head and relation, while the other only composed of words from tail. Then, the representations for all possible combinations of 'head + relation', as well as the representation of tail, are computed and stored in advance by feeding them into the BERT encoder. When a triplet is given for testing, we just need to retrieve the representations corresponding to its two segments from the storage. Despite lots of time being saved, we find that the performance gap between bidirectional prediction is aggravated. For instances, we observe that StAR is easier to predict the missing entity 'Jobs' in $(?, Founding, Apple)$, while struggling in giving a correct prediction for $(Jobs, Founding, ?)$. Intuitively, the performance gap between the head-to-tail and tail-to-head prediction should not be aggravated since they convey the same semantic information. By examining the model, we think the unbalanced performance between two directions may partially come from the asymmetric encoder structure and triplet splitting, which gives the head entity an unreasonable priority over the tail by always associating the relation with it.

To alleviate the issue, we developed a model to balance the roles played by the head and tail entities via adding another dual encoder into the original StAR model. By associating the relation with head and tail entities, respectively, two different splittings are obtained for every triplet, that is, (head + relation, tail) and (head, relation + tail). By feeding the two splittings into the two dual encoder branches of our model, respectively, we obtain two representations for every triplet, with each giving priority to the head and tail. Since the two representations arise from the same triplet, they should contain the same semantic information, but the position-sensitive BERT cannot guarantee this. To align semantics between the two representations, a semantic alignment mechanism is further proposed based on contrastive learning, which has proven its effectiveness in extracting the semantic information in both images and texts. To further improve the prediction performance, a hard negative sampling technique is also proposed to train the model, with a theoretical analysis provided to explain its effectiveness. We evaluated our model on three public datasets WN18RR, FB15K-237 and UMLS, and significant performance improvements have been observed over comparable baselines.

2 Preliminaries

Knowledge Graph Completion A knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$ is a collection of triplets (h, r, t) that present commonsense relations between pairs of entities, where $h, t \in \mathcal{R}$ are the head and tail entities respectively and $r \in \mathcal{R}$ represents the relation between them. Given a head h (or tail t) entity and a relation, the task of knowledge graph completion (KGC) aims at predicting the most possible tail t (or head h) to make the new triplet (h, r, t) plausible in $\mathcal{G}$. Specifically, given a incomplete triplet $(h, r, ?)$, the model seeks the best-suited tail entity by enumerating every entity in $\mathcal{E}$ and calculates a score function $f_{\boldsymbol{\theta}} : \mathcal{G} \rightarrow \mathbb{R}$ to gauge its suitability. The final triplet is completed by adding the tail entity via $t' = \text{argmax}_{t \in \mathcal{E}} f(h, r, t)$.

Text-based Knowledge Graph Completion Completing knowledge graphs is challenging, since the model needs to discover the commonsense implied in the triplets. The current prevailling works mainly focus on how to learn a meaningful contextualized representation for triplets. KG-BERT first applies pre-trained language models to learn such informative representations, while suffers from burdened computational cost during inference. StAR circumvents this problem by using a separate encoding pattern. Specifically, denoting the text representation of a triplet (h, r, t) as $(x^{(h)}, x^{(r)}, x^{(t)})$, instead of that done in KG-BERT, which takes a complete triplet as input, StAR first constructs two types of descriptions for the triplet via:

$$\begin{aligned} H_h &= [\langle \text{CLS} \rangle, x^{(h)}, \langle \text{SEP} \rangle, x^{(r)}, \langle \text{SEP} \rangle], \\ H_t &= [\langle \text{CLS} \rangle, x^{(t)}, \langle \text{SEP} \rangle], \end{aligned} \tag{1}$$

where $\langle \text{CLS} \rangle$ and $\langle \text{SEP} \rangle$ are the special token for classification and sentence separation in the Tansformer architecture, respectively. Then the corresponding contextualized representations are encoded by a Transformer encoder

$$\begin{aligned} \boldsymbol{u}_h &= \text{Transformer-Enc}(H_h)[0], \\ \boldsymbol{u}_t &= \text{Transformer-Enc}(H_t)[0], \end{aligned} \tag{2}$$

where index 0 stands for the position of $\langle \text{CLS} \rangle$'s embedding. Finally, the structure-aware represen-

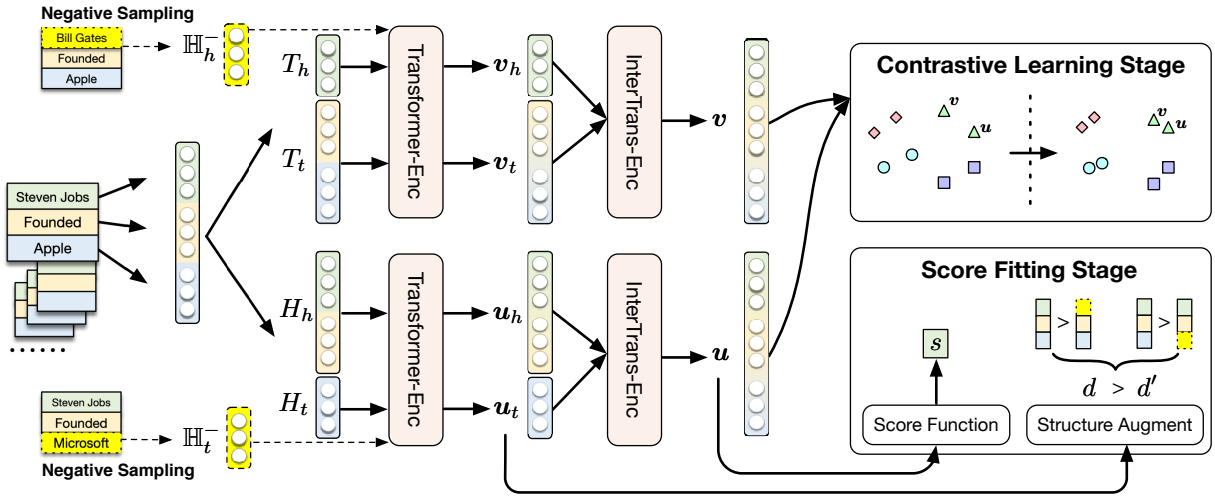

Figure 2: Illustration of our asymmetry alleviation framework.

tation is obtained by a interactive concatenation

$$\boldsymbol{u} = [\boldsymbol{u}_h; \boldsymbol{u}_h \times \boldsymbol{u}_t; \boldsymbol{u}_h - \boldsymbol{u}_t; \boldsymbol{u}_t]. \quad (3)$$

We denote this Transformer-based interactive encoder function as

$$\boldsymbol{u} = \text{InterTrans-Enc}(H_h, H_t; \boldsymbol{\theta}), \quad (4)$$

where $\boldsymbol{\theta}$ denotes the parameters. Using such a separate encoding framework, we can store the encoded representations in advance, and then obtain the representation of a new triplet via the concatenation operator in (3), which would significantly improve the inference efficiency. Besides, as mentioned in (Wang et al., 2021), the embedding encoded in such framework is informative enough for trained score function to distinguish a true/false triple relation, and yields appealing performance gains.

3 The Asymmetry Alleviation Framework

In this section, we introduce our model architecture illustrated in Figure 2. We first point out the limitations of StAR, and then propose dual Transformer encoders with semantics alignment mechanism to alleviate this limitation.

3.1 Limitations of StAR

Although StAR achieves considerable performance, we observe in the experiments (see Table 4) that it is generally good at unidirectional prediction, but performs worse at reverse prediction. Take the UMLS dataset as example, the model works well in head-to-tail prediction, *i.e.*, $(?, r, t) \rightarrow h$. However, in tail-to-head prediction, *i.e.*, $(h, r, ?) \rightarrow t$, its performance declines shapely. By thoroughly analysing the architecture of StAR, we find that the potential hazard arose at the asymmetric encoding manner. As shown in (1), the relation solely interacts with the head entity, but totally ignores the tail. Such asymmetric interaction patterns might encourage the model to outweigh the importance of head entity and exclusively focus on unidirectional prediction. Motivated by our findings, we seek for methods that augment the StAR model to alleviate the performance gap between bidirectional predictions.

3.2 Dual Transformer Encoders

To alleviate the asymmetric problem, the output of encoder should be irrelevant with the pattern of how triplets are concatenated as input. That is, we expect the encoder to give equal attention to head and tail entities, rather than just giving a high priority to one by associating it with a relation but neglecting the other. In this end, we propose a symmetric architecture to generate a direction insensitive triplet representation. Specifically, in addition to generate a heavy-head representation as defined in (4), we also require the encoder to output a heavy-tail representation, by first constructing the descriptions as

$$\begin{aligned} T_h &= [\langle \text{CLS} \rangle, x^{(h)}, \langle \text{SEP} \rangle], \\ T_t &= [\langle \text{CLS} \rangle, x^{(r)}, \langle \text{SEP} \rangle, x^{(t)}, \langle \text{SEP} \rangle]. \end{aligned} \quad (5)$$

Compared (5) with (1), it can be observed that the difference lies at whether the relation is concatenated with head or tail entity. By doing so, we can enforce the encoder to balance the status of head and tail entities with respect to the relation, partially alleviating the asymmetric problem to some extend.

Then the heavy-tail representation is generated by the same interactive pattern of (4)

$$\boldsymbol{v} = \text{InterTrans-Enc}(T_h, T_t; \boldsymbol{\theta}). \quad (6)$$

Under our assumption, the heavy-tail representation $\boldsymbol{v}$ should maintain similar semantic information with the heavy-head representation $\boldsymbol{u}$, since both of them are originated from the same triplet relation. However, the vanilla Transformer encoder can not guarantee this deserved property, due to its natural sensitivity of the order of input sequences. Thereby, an additional mechanism is required to make the two types of representations share homologous information.

3.3 Semantic Alignment Mechanism

To align the semantic space of $\boldsymbol{u}$ and $\boldsymbol{v}$, we can first propose a distance measure to judge their semantic difference, and then minimize the corresponding gap. There are a lot of distance measures can be used to achieve this goal, such as L_2 distance and cosine similarity. However, such a simple distance measure works unsatisfactory, as shown in Table 5. The reason why it malfunctions is apparent. Both L_2 and cosine measures are only responsible for attracting the representations $\boldsymbol{u}$ and $\boldsymbol{v}$ closer, while not taking into account the preservation of their semantic information at all. With these alignment measures, the representations are all inclined to collapse into a single point, discarding all the meaningful semantic information contained in the triplets.

To align the two representations while preserving their semantic information, regarding the two types of concatenations as the views of a triplet relation, we find that the proposed dual Transformer encoder is similar to the model architecture of contrastive learning (Chen et al., 2020), which generates two views of a given image by some random operators, *e.g.*, rotation, cropping, resizing, etc. This inspires us to align the semantics of $\boldsymbol{u}$ and $\boldsymbol{v}$ by using contrastive loss. Particularly, denoting the representations output by the dual Transformer encoders as $\mathcal{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_b\}$ and $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_b\}$, then the NT-Xent contrastive loss (Chen et al., 2020) is defined as

$$\mathcal{L}_{\text{CL}} = -\sum_i \log \frac{\exp(\boldsymbol{u}_i^T \boldsymbol{v}_i/\tau)}{\sum_{\boldsymbol{n}\in\mathcal{U}\cup\mathcal{V}\setminus\boldsymbol{u}_i} \exp(\boldsymbol{u}_i^T \boldsymbol{n}/\tau)}, \quad (7)$$

where b denotes the batch size and τ is the temperature coefficient. By minimizing $\mathcal{L}_{\text{CL}}$, we can impose the representations of two views (*i.e.,* $\boldsymbol{u}$ and $\boldsymbol{v}$) in a triplet relation to be closer, while stay away from the others in semantic space, as illustrated in Figure 2. This appealing property of contrastive objective has been widely applied in image representation learning (Wang and Isola, 2020) and cluttering (Li et al., 2021). Here, we find that it also generates more expressive representations of triplet relations, which can significantly improve the performance of knowledge graph completion and benefit to alleviate the asymmetry problem.

Datasets	Candidates	Hit@1↑	Hit@3↑	Hit@10↑	MMR↑	MR↓
WN18RR	normal	.237	.508	.725	.405	47.000
	exclude	**.712**	**.852**	**.930**	**.793**	**21.800**
FB15k-237	normal	.193	.316	.481	.288	115.830
	exclude	**.512**	**.664**	**.778**	**.607**	**42.420**
UMLS	normal	.793	.964	.992	.881	1.508
	exclude	**.952**	**.981**	**1.000**	**.969**	**1.224**

Table 1: The impact of excluding hard candidates in knowledge graph complete task.

4 Further Improving by Training with Hard Negative Sampling Strategy

To obtain the final score of a given triplet, we can use a function applied to the generated representations

$$s = \exp(\text{MLP}([\boldsymbol{u}, \boldsymbol{v}]; \boldsymbol{\psi})),$$

where σ denotes the sigmoid function and $\text{MLP}(\cdot)$ stands for a multi-layer perceptron. For brevity, we denote the overall model as a score function $s = f_{\boldsymbol{\theta},\boldsymbol{\psi}}(h, r, t)$, where $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$ denote the parameters of Transformer encoder and MLP, respectively. To train model, we can encourage it to give the true triplets high scores, while low scores for false (fabricated) triplets. In particular, the model can be trained by minimizing the following score fitting loss

$$\mathcal{L}_{\text{SF}} = -\mathbb{E}_{q(\phi^+)}[\log f(\phi^+)] + \mathbb{E}_{q(\phi^-)}[\log f(\phi^-)], \quad (8)$$

where $q(\phi^+)$ and $q(\phi^-)$ denote the distribution of true and false triplets. In practice, the set of false triplets are obtained as $\mathbb{D}^- = \mathbb{D}_h^- \cup \mathbb{D}_t^-$, with the definition of

$$\mathbb{D}_h^- = \left\{(h', r, t) \middle| h' \in \mathcal{E} \wedge (h', r, t) \notin \mathbb{D}^+\right\},$$
$$\mathbb{D}_t^- = \left\{(h, r, t') \middle| t' \in \mathcal{E} \wedge (h, r, t') \notin \mathbb{D}^+\right\},$$

and $\mathbb{D}^+$ denotes the set of true triplets. However, we find that the model trained with the proposed

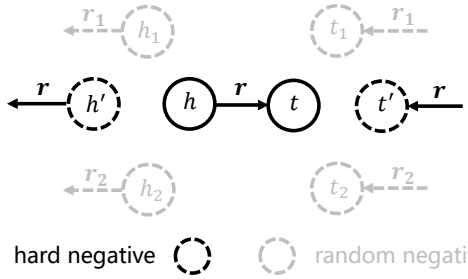

Figure 3: Illustration of hard negative sampling strategy.

false triplet set can not distinguish hard samples well. In particular, given a test triplet, if we exclude out all entities connected by the same relation in the candidates set except the correct one, the prediction accuracy can be improved significantly, as shown in Table 1. This inspires us that additional performance gains can be obtained if we explicitly tell the model which are hard samples and pay more attention to them during the training. Motivated by this finding, next we propose to exploit the hard samples to further improve the prediction performance.

4.1 Hard Negative Sampling Strategy

Basically, we want the model to give more emphasis on the samples that are more likely to be predicted incorrectly. However, the problem is how to identify these so-called hard samples. To identify the hard negative samples for a triplet (h, r, t), instead of defining the support set of $q(\phi^-)$ over $\mathbb{D}^-$, we restrict it in a hard negative domain $\mathbb{H}^-$. For easier understanding here, we only demonstrate the sampling procedures of the hard negative samples, whereas the rigorous definition of $\mathbb{H}^-$ is available in the appendix A.1. In particular, the hard negative samples of (h, r, t) can be sampled from $\mathbb{H}^-$ via the following three steps: i) search for a new triplet (h', r, t') obeyed the relation r; ii) replace the head and tail entity with h' and t' to obtain (h', r, t) and (h, r, t'); and iii) check whether they lies in $\mathbb{D}^-$, if not, go back to step i, else accept them as hard negative samples. We illustrate this strategy in Figure 3 for further comprehension.

Understanding the Hard Negative Sampling Strategy In order to understand the impact of hard negative samples,, we can construct an energy-based model with the form

$$p(h, r, t) = \frac{f(h, r, t)}{Z}, \tag{9}$$

Algorithm 1 Model Training Algorithm

Input: KG's positive triplets set $\mathbb{D}^+$; Negative triplets set $\mathbb{H}^-$; batch size b; number of negative samples n;
Output: Optimal parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat ▷ Contrastive learning stage
  $\mathcal{P} \leftarrow \{\phi_1^+, \ldots, \phi_b^+\} \sim \mathbb{D}^+$ ▷ Sample positive triplets
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CL}}(\mathcal{P}; \boldsymbol{\theta})$
  $\boldsymbol{\theta} \leftarrow$ Update parameters using $\boldsymbol{g}$ (*i.e.,* Adam)
until convergence of parameters $(\boldsymbol{\theta})$
repeat ▷ Score fitting stage
  $\mathcal{P} \leftarrow \{\phi_1^+, \ldots, \phi_b^+\} \sim \mathbb{D}^+$ ▷ Sample positive triplets
  $\mathcal{N} \leftarrow \{\phi_1^-, \ldots, \phi_n^-\} \sim \mathbb{H}^-$ ▷ Sample negative triplets
  $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}, \boldsymbol{\psi}} \mathcal{L}_{\mathrm{SF}}(\mathcal{P}, \mathcal{N}; \boldsymbol{\theta}, \boldsymbol{\psi}) + \mathcal{L}_{\mathrm{SA}}(\mathcal{P}, \mathcal{N}; \boldsymbol{\theta}, \boldsymbol{\psi})$
  $\boldsymbol{\theta}, \boldsymbol{\psi} \leftarrow$ Update parameters using $\boldsymbol{g}$ (*i.e.,* Adam)
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\psi})$

where $Z = \sum_{h' \neq h} f(h', r, t) + \sum_{t' \neq t} f(h, r, t'))$ denotes the partition function. It can be turned out that the objective $-\mathcal{L}_{\mathrm{SF}}$ in (8) is the lower bound of $\log p(h, r, t)$. That means minimizing $\mathcal{L}_{\mathrm{SF}}$ is equivalent to maximize the log likelihood $\log p(h, r, t)$. Moreover, using hard negative samples in (8) can achieve a tighter bound than using random negative samples. Thus, the hard negative sampling strategy would result in better local optimal after training. We summarize the above observation in the following proposition.

Proposition 4.1. *Define a probabilistic density* $p = \frac{f(h,r,t)}{Z}$ *with* $Z = \sum_{h' \neq h} f(h', r, t) + \sum_{t' \neq t} f(h, r, t'))$, *we have* $\operatorname{argmin}_{\boldsymbol{\theta}, \boldsymbol{\psi}} \mathcal{L}_{\mathrm{SF}} \Leftrightarrow \operatorname{argmax}_{\boldsymbol{\theta}, \boldsymbol{\psi}} \log p$, *more importantly,* $\log p \geq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{H}^-}) \geq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{D}^-})$.

Proof. Please refer to the appendix A.2. □

4.2 The Two-stage Training

To train the model, we can simultaneously optimize the contrastive learning and score fitting objective. Besides, the same as StAR, we further enforce the distance between contextualized representations in (2) to be closed. Specifically, we define a structure distance as $d = -\|\boldsymbol{u}_h - \boldsymbol{u}_t\|^2$, then the the distance can be shrinked by minimizing the ranking loss

$$\mathcal{L}_{\mathrm{R}}^{\boldsymbol{u}} = \max(0, \lambda - d + d'), \tag{10}$$

where λ is the margin and d' denotes the distance of a negative sample. This objective is known as structure augmentation, which can reduce disambiguating entities and push model to produce more reliable ranking scores (Wang et al., 2021). Similarly, we can also design a ranking loss $\mathcal{L}_{\mathrm{R}}^{\boldsymbol{v}}$ for the heavy-tail representation in (6). Then the structure augmentation objective is

$$\mathcal{L}_{\mathrm{SA}} = \mathcal{L}_{\mathrm{R}}^{\boldsymbol{u}} + \mathcal{L}_{\mathrm{R}}^{\boldsymbol{v}}. \tag{11}$$

Finally, we obtain three training objectives: contrastive loss $\mathcal{L}_{\mathrm{CL}}$, score fitting $\mathcal{L}_{\mathrm{SF}}$ and structure augmentation $\mathcal{L}_{\mathrm{SA}}$. Instead of optimizing all of them combinatorially, we propose a two-stage training algorithm. Specifically, we first minimize $\mathcal{L}_{\mathrm{CL}}$ in the contrastive learning stage, and then optimize $\mathcal{L}_{\mathrm{SF}} + \mathcal{L}_{\mathrm{SA}}$ in the score fitting stage. The details are shown in Algorithm 1. After training, given an incomplete triplet $(h, r, ?)$, we can predict the corresponding tail entity via $t = \mathrm{argmax}_{t'} f(h, r, t')$. The tail-to-head prediction can be done in the same way.

5 Related Work

KGC contains methods based on graph embedding and methods based on text embedding. The distance-based translation model evaluates the confidence of the triplet by applying a translation function to the embeddings of head and relation and selecting the closest tail as the result, the most typical are TransE (Bordes et al., 2013) and RotatE (Sun et al., 2019). Semantic matching models, such as RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2015) and QutatE (Zhang et al., 2019), the score function of them usually are defined as a bilinear function. Deep neural network methods pay more attention to the interaction between entities and relationships, that is particularly obvious in the ConvE (Dettmers et al., 2018) based on convolutional neural networks (CNN) (Kipf and Welling, 2017). The Graph Convolutional Neural network (GCN) (Dettmers et al., 2018) and the Attention Graph Network (GAT) (Velickovic et al., 2018) in the graph neural network have also been introduced into KGC. The most typical ones are the GCN-based SACN (Shang et al., 2019) and the GAT-based KBAT (Nathani et al., 2019). However, All graph-based models mentioned above are greatly limited by the graph structure and are not suitable for large-scale dynamically changing KGs at all.

In order to get rid of the limitations of graph structure and make full use of context information in KG, KGC models based on text embedding are proposed, represented by KG-BERT (Yao et al., 2019) and StAR (Wang et al., 2021). KG-BERT expresses entities and relationships as their names or descriptions, and then uses the name or the word sequence of the description as the input sentences of the fine-tuned BERT model. KG-BERT shows good performance, but one of the main limitations of KG-BERT is that its extremely high cost in evaluate and predict stages. In order to solve this problem, StAR introduced two-branch Siamese architecture (Zagoruyko and Komodakis, 2015), which greatly improved the inference speed. StAR has surpassed KG-BERT in model reasoning speed and experimental effect. However, the prediction results of StAR show serious asymmetry, which contradicts the independence and completeness of a text representing a triplet. Our work developed a model to balance the roles played by the head and tail entities via adding another dual encoder into the original StAR model, and alleviated the asymmetry problem in KGC.

Dataset	Entity	Relation	Train	Dev	Test
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20446
UMLS	135	46	5216	652	661

Table 2: Summary statistics of datasets.

6 Experiment

6.1 Experiment Setup

Datasets We make evaluations on three public KG datasets with the summary statistics shown in Table 2. i) WN18RR: a link prediction dataset created from the WordNet, which is ensured that the evaluation dataset does not have inverse relation test leakage (Dettmers et al., 2018); ii) FB15k-237: a set of Freebase entity pairs which contains knowledge base relation triplets and textual mentions, in which the inverse relations of original dataset are removed (Xie et al., 2016); iii) UMLS: a compendium of many controlled vocabularies in the biomedical sciences that consists of knowledge sources and a set of software tools.

Baselines We compare our model with several state-of-the-art models:, covering the latest models that are the best in terms of various indicators. These include TransE (Bordes et al., 2013), DisMult (Yang et al., 2015), CompIEx (Trouillon et al., 2016), KBGAN (Cai and Wang, 2018), R-GCN (Schlichtkrull et al., 2018), ConvE (Dettmers et al., 2018), convKB (Nguyen et al., 2018), KBAT (Nathani et al., 2019), CapsE (Nguyen et al., 2019), QuatE (Zhang et al., 2019), RotatE (Sun et al., 2019), TuckER (Balazevic et al., 2019), AttH (Chami et al., 2020), KG-BERT (Yao et al., 2019), and StAR (Wang et al., 2021).

Models	WN18RR					FB15k-237					UMLS				
	MR↓	MRR↑	Hits↑			MR↓	MRR↑	Hits↑			MR↓	MRR↑	Hits↑		
			@ 1	@ 3	@ 10			@ 1	@ 3	@ 10			@ 1	@ 3	@ 10
Graph Embedding Approch															
TransE△	2300	.243	.043	.441	.532	323	.279	.198	.376	.441	1.840	-	-	-	.989
DisMult△	7000	.444	.412	.470	.504	512	.281	.199	.301	.446	5.520	-	-	-	.846
CompIEx△	7882	.449	.409	.469	.530	546	.278	.194	.297	.450	2.590	-	-	-	.967
KBGAN	-	.215	-	-	.469	-	.277	-	-	.458	-	-	-	-	-
R-GCN△	6700	.123	.080	.137	.207	600	.164	.100	.181	.300	-	-	-	-	-
ConvE△	4464	.456	.419	.470	.531	245	.312	.225	.341	.497	**1.510**	-	-	-	**.990**
ConvKB◊	3433	.249	-	-	.524	309	.243	-	-	.421	-	-	-	-	-
KBAT◊	1921	.412	-	-	.554	270	.157	-	-	.331	-	-	-	-	-
CapsE◊	**718**	.415	-	-	.559	403	.150	-	-	.356	-	-	-	-	-
QuatE	3472	.481	.436	**.500**	.564	**176**	.311	.221	.342	.495	-	-	-	-	-
RotatE	3340	.476	.428	.492	.571	177	.338	.241	.375	.533	-	-	-	-	-
TuckER	-	.470	**.443***	.482	.526	-	**.358***	**.266***	**.394***	**.544***	-	-	-	-	-
AttH	-	**.486***	**.443***	.499	**.573**	-	.348	.252	.384	.540	-	-	-	-	-
Text Encoding Approch															
KG-BERT	97	.216	.041	.302	.524	153	-	-	-	.420	1.550	.870	.790	.937	.990
StAR	**54***	.411	**.264**	.491	.709	**117***	.288	.193	.315	.481	1.970	.836	.729	.933	.991
Ours	**54***	**.412**	.235	**.517***	**.750***	119	**.291**	**.197**	**.321**	**.484**	**1.508***	**.881***	**.793***	**.964***	**.992***

Table 3: Comparison of the proposed method against baseline models. △ marked results are reported by (Nathani et al., 2019), ◊ marked results are re-evaluated by (Sun et al., 2020), StAR is re-implemented by ourselves using its public code and the others are taken from the original papers. Best results for each genre are marked in bold and the started numbers denote the state-of-the-art performance. ↑ means higher is better, and ↓ versa.

Datasets	Method	Hit@1↑			Hit@3↑			Hit@10↑			MMR↑			MR↓		
		left	right	diff	left	right	diff	left	right	diff	left	right	diff	left	right	diff
WN18RR	star	.231	.297	.066	.456	.526	.070	.665	.751	.086	.368	.449	.081	57.549	51.098	6.451
	ours	.208	.264	**.056**	.483	.551	**.068**	.712	.783	**.071**	.382	.441	**.059**	53.206	49.194	**4.012**
FB15K-237	star	.111	.275	.164	.218	.413	.195	.389	.573	.184	.201	.375	.174	140.266	93.734	46.532
	ours	.121	.273	**.152**	.229	.413	**.184**	.394	.573	**.179**	.218	.376	**.158**	135.037	94.965	**40.072**
UMLS	star	.770	.689	.081	.969	.897	.072	.993	.968	.025	.870	.802	.068	1.459	2.481	1.022
	ours	.809	.778	**.031**	.982	.946	**.036**	.997	.986	**.011**	.897	.865	**.032**	1.298	1.718	**.420**

Table 4: Comparison of the asymmetry alleviation degree on the UMLS dataset. "Left" means the tail-to-head prediction, *i.e.*, $(?, r, t) \rightarrow h$, "right" means the head-to-tail prediction, *i.e.*, $(h, r, ?) \rightarrow t$ and "diff" means the performance difference of bidirectional predictions.

Evaluation Protocol For the sake of fairness, we follow the evaluation protocol in (Sun et al., 2020). Particularly, given a test triplet, we first corrupt its head or tail using other entities in the KG. Then the trained model ranks ground triplet over the corrupted ones according to their scores with "filtered" setting (Bordes et al., 2013). we adopt three metrics: Mean Rank (MR), Mean Reciprocal Rank (MRR) and Hits at N (Hit@N with $N = 1, 3,$ 10) as evaluation criteria separately.

Implementation Details In our experiment, the Transformer encoder is a 12-layer, 12-head, 768-dimensional RoBERTa module (Liu et al., 2019) initialized from its public pre-trained parameters. The input text data is a truncated word sequence with the length of $32, 128, 16$ for the WN18RR, FB15K-237 and UMLS, respectively. The temperature coefficient of contrastive loss in (7), the number of negative samples in (8) and the margin coefficient in (10) are fixed to 0.05, 1 and 5, respectively. We train the model using Adam optimizer (Kingma and Ba, 2015) with the learning rate of $2e-5$ and the dropout rate of 0.1. According to the performance observed on the validation set, we choose the batch size from $\{16, 32\}$ and the epochs for each training stage from $\{5, 10, 20\}$, with the best best setting used for evaluation on the test set.

6.2 Performance and Analysis

Main Results The performance of our model and competitive baselines are illustrated in Table 3. In can be observed that the proposed model achieves superior performance on all datasets and surpasses state-of-art models on most metrics. Especially in terms of MR, which is an extremly important

Distance	Hit@1↑	Hit@3↑	Hit@10↑	MRR↑	MR↓
L_2	0.397	0.663	0.836	0.561	6.547
Cosine	0.437	0.674	0.856	0.583	6.691
CL	**0.793**	**0.964**	**0.992**	**0.881**	**1.508**

Table 5: Effect of using different semantic alignment strategies on the UMLS dataset.

	Method	Hit@1↑	Hit@3↑	Hit@10↑	MRR↑	MR↓
1	Random	.713	.899	.975	.818	1.941
	Hard	**.719**	**.903**	**.984**	**.829**	**1.830**
3	Random	.781	.947	.980	.852	1.613
	Hard	**.794**	**.952**	**.989**	**.876**	**1.549**
5	Random	.789	.951	.991	.863	1.512
	Hard	**.793**	**.964**	**.992**	**.881**	**1.508**

Table 6: Comparison of using random and hard negative samples on the UMLS dataset. $(1, 3, 5)$ means the number of negative samples.

indicator in information retrieval, the significant performance gains obtained in this metric demonstrate the superiority of our model, indicating that better semantic information can be extracted by our model. Besides, on UMLS dataset, our model consistently outperforms baselines on all metrics, and on WN18RR, our model surpasses all other methods by a large margin in terms of Hits@10. These remarkable gratuities strongly corroborate the benefit of our contrastive-based asymmetry alleviation framework. When examining the performance on the FB15k-237 dataset, we find that all text-based approaches inferior than graph-based ones. This phenomenon is consistent with experiment results in previous works (Wang et al., 2021). In spite of that, thanks to the introduction of asymmetry alleviation mechanism, our model still remarkably outperforms other test embedding methods, like KG-BERT and StAR.

Asymmetry Alleviation Analysis The asymmetry issue is manifested in that, for the same test triplet, the effect of prediction from head to tail, *i.e.,* $(h, r, ?) \rightarrow t$, and prediction from tail to head, *i.e.,* $(?, r, t) \rightarrow h$, varies significantly. To test whether our model can alleviate this problem, we explicitly carry out experiments in the two predicted directions. The statistical results on all metrics and datasets are shown in Table 4. It can be seen that, compared with StAR, our model achieves smaller performance gap between predictions in two directions. Especially in the UMLS dataset, our model reduces the performance gap by a factor of two. This result clearly shows that our model alleviates the problem of asymmetry in prediction to a certain extent, and thus shows better performance overall.

Impact of Semantics Alignment The semantic alignment mechanism plays an important role in our model architecture. To investigate the influence of different alignment methods, we further experiment with two different distance measures: the L_2 distance and cosine distance. As seen form Table 5, contrastive loss achieves better performance than the other two measures. This is partially because, the L_2 and cosine distance only impose the two type of representations (it i.e., $\boldsymbol{u}$ and $\boldsymbol{v}$) to be closed, but not push them away from the negative samples, which is inclined to make all representations collapse in a single compact hyperspace and result in indistinguishable and meaningless embeddings. However, by simultaneously requiring the alignment and uniformity of the generated features (Wang and Isola, 2020), contrastive loss successfully aligns the semantics of two types of representations and thus improves the performance.

Effect of Hard Negative Sampling To evaluate the effect of the hard negative sampling strategy, we compare the model performance of using random and hard negative samples. As shown in Table 6, under the premise of the same number of negative samples, the performance of using hard negative samples are better than that of using random samples. Besides, we find that increasing the number of negative samples could introduce additional performance gains. However, large number would significantly improve the computational complexity. Based on the trade-off between computational cost and performance benefits, we find 5 is a suitable number for negative sampling.

7 Conclusion

We proposed a method for knowledge graph completion. Specifically, we introduced a dual Transformer encoding framework combined with a semantics alignment mechanism to alleviate the asymmetry problem of StAR. To enhance model ability, a negative sampling strategy was further developed and justified theoretically. Extensive evaluations demonstrated that our model significantly outperforms baseline methods and effectively reduces the performance gaps of bidirectional prediction.

References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5185–5194, Hong Kong, China. Association for Computational Linguistics.

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.

Liwei Cai and William Yang Wang. 2018. KBGAN: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1470–1480, New Orleans, Louisiana. Association for Computational Linguistics.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6901–6914, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yang Gao, Yi-Fan Li, Yu Lin, Hang Gao, and Latifur Khan. 2020. Deep learning on knowledge graph for recommender system: A survey. *ArXiv preprint*, abs/2004.00387.

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 105–113. ACM.

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*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. 2021. Prototypical contrastive learning of unsupervised representations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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. *ArXiv preprint*, abs/1907.11692.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy. Association for Computational Linguistics.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333, New Orleans, Louisiana. Association for Computational Linguistics.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189, Minneapolis, Minnesota. Association for Computational Linguistics.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 809–816. Omnipress.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3060–3067. AAAI Press.

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 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*, pages 5516–5522, Online. Association for Computational Linguistics.

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.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, pages 1737–1748.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2659–2665. AAAI Press.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1271–1279. ACM.

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. *ArXiv preprint*, abs/1909.03193.

Sergey Zagoruyko and Nikos Komodakis. 2015. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4353–4361. IEEE Computer Society.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. 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 2731–2741.

A Appendix

A.1 Definition of $\mathbb{H}^-$

In section 4.1, we propose to draw the negative samples form a hard domain $\mathbb{H}^-$. Here, we provide the rigid definition of $\mathbb{H}^-$. Specifically, given a triplet (h,r,t), its hard negative samples can be drew from the set $\mathbb{H}^- = \mathbb{H}_h^- \cup \mathbb{H}_t^-$, with the definition of

$$\mathbb{H}_h^- = \left\{(h',r,t) \middle| h' \in \mathcal{E} \wedge (h',r,t) \in \mathbb{D}_h^- \cap \mathbb{T}_h^+\right\},$$
$$\mathbb{H}_t^- = \left\{(h,r,t') \middle| t' \in \mathcal{E} \wedge (h,r,t') \in \mathbb{D}_t^- \cap \mathbb{T}_t^+\right\},$$

where $\mathbb{T}_h^+$ and $\mathbb{T}_t^+$ are respectively defined as

$$\mathbb{T}_h^+ = \left\{(h',r,t) \middle| \exists t' \in \mathcal{E}, h' \in \mathcal{E} \wedge (h',r,t') \in \mathbb{D}^+\right\},$$
$$\mathbb{T}_t^+ = \left\{(h,r,t') \middle| \exists h' \in \mathcal{E}, t' \in \mathcal{E} \wedge (h',r,t') \in \mathbb{D}^+\right\}.$$

Roughly speaking, the hard negative sample for a given triplet (h,r,t) is defined as a corrupt triplet $(h',r,t) \in \mathbb{D}^-$ (or $(h,r,t') \in \mathbb{D}^-$), in which the substituted entity h' (or t') should be related to one entity in the knowledge graph $\mathcal{G}$ with relation r.

A.2 Proof of Proposition 4.1

To train our model $f(h,r,t)$, we can define an energy-based model as

$$p(h,r,t) = \frac{f(h,r,t)}{Z},$$

where $Z = \sum_{h' \neq h} f(h',r,t) + \sum_{t' \neq t} f(h,r,t'))$ denotes the partition function, and then maximize the log likelihood of $p(h,r,t)$. Next, we show that the training objective in (8) is equivalent to the maximum likelihood estimation.

To facilitate discussion, we denote $\phi \triangleq (h,r,t)$ in the following. We have

$$\begin{aligned}\log p(\phi) &= \log f(\phi) - \log Z \\ &= \log f(\phi) - \mathbb{E}_{q(\phi)}[\log f(\phi)] \\ &\quad - H(q(\phi)) - KL(q||p) \\ &\geq \log f(\phi) - \mathbb{E}_{q(\phi)}[\log f(\phi)], \qquad (12)\end{aligned}$$

where $H(\cdot)$ denotes the Shannon entropy and $KL(\cdot||\cdot)$ is the Kullback–Leibler divergence. The last inequality holds since the entropy and KL divergence are both non-negative for discrete distribution. By constricting the support set of $q(\phi)$ to be $\mathbb{D}^-$, we have

$$\underset{\boldsymbol{\theta},\boldsymbol{\psi}}{\operatorname{argmax}} \log p \Leftrightarrow \underset{\boldsymbol{\theta},\boldsymbol{\psi}}{\operatorname{argmin}} \mathcal{L}_{\mathrm{SF}}.$$

Next, we show that the inequality in Proposition 4.1 holds. We first denote $q(\phi^-)$ with the support set on $\mathbb{D}^-$ as $q_{\mathbb{D}^-}(\phi)$, and $q_{\mathbb{H}^-}(\phi)$ denotes the negative distribution over $\mathbb{H}^-$. Then according to the (12), we have

$$\log p(\phi) \geq \log f(\phi) - \mathbb{E}_{q_{\mathbb{H}^-}}[\log f(\phi)] \triangleq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{H}^-}).$$

w.l.t.g., since $q_{\mathbb{D}^-}(\phi)$ is more widely uniform than $q_{\mathbb{H}^-}(\phi)$, thereby $H(q_{\mathbb{D}^-}) \geq H(q_{\mathbb{H}^-})$. Besides, since the triplet in $\mathbb{H}^-$ is harder than that in $\mathbb{D}^-$, it is reasonable to assume the score of entities in $\mathbb{H}^-$ is higher than those in $\mathbb{D}^-$. So we have $KL(q_{\mathbb{H}^-}||p) \leq KL(q_{\mathbb{D}^-}||p)$. According to (12), the following inequality holds

$$\begin{aligned}&-\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{D}^-}) \triangleq \log f(\phi) - \mathbb{E}_{q_{\mathbb{D}^-}}[\log f(\phi)] \\ &\leq \log f(\phi) - \mathbb{E}_{q_{\mathbb{H}^-}}[\log f(\phi)] \triangleq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{H}^-}).\end{aligned}$$

Thus, we end up with our proof at

$$\log p \geq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{H}^-}) \geq -\mathcal{L}_{\mathrm{SF}}(q_{\mathbb{D}^-}).$$

This inequality shows that, compared with the random negatives, the hard negative sampling strategy has a tighter lower bound and thus achieves a better estimation of the likelihood.

A.3 Case Study

In Table 7, we present some ranking cases of the given test triplets on the UMLS dataset. It can be seen that our model works well in both head-to-tail and tail-to-head predictions. However, StAR is generally good at unidirectional prediction but performs worse at reverse prediction. This observation further confirms the effectiveness of our asymmetry alleviation mechanism.

Test triplet	Predict	StAR	Ours
(receptor, isa, substance)	tail-to-head	**1. receptor** 2. body_part_organ_or_organ_component 3. medical_device 4. regulation_or_law	**1. receptor** 2. medical_device 3. drug_delivery_device 4. clinical_drug
	head-to-tail	1. mental_process 2. immunologic_factor 3. vitamin **9. substance**	**1. substance** 2. chemical_viewed_structurally 3. indicator_reagent_or_diagnostic_aid 4. organic_chemical
(cell, location_of, therapeutic_or_preventive_procedure)	tail-to-head	**1. cell** 2. gene_or_genome 3. fully_formed_anatomical_structure 4. embryonic_structure	**1. cell** 2. gene_or_genome 3. fully_formed_anatomical_structure 4. embryonic_structure
	head-to-tail	1. natural_phenomenon_or_process 2. clinical_attribute 3. organism_attribute **5. therapeutic_or_preventive_procedure**	1. natural_phenomenon_or_process **2. therapeutic_or_preventive_procedure** 3. alga 4. fish
(steroid, interacts_with, eicosanoid)	tail-to-head	1. chemical_viewed_functionally 2. element_ion_or_isotope 3. inorganic_chemical **4. steroid**	1. eicosanoid **2. steroid** 3. chemical_viewed_functionally 4.inorganic_chemical
	head-to-tail	1. body_substance 2. lipid 3. biomedical_occupation_or_discipline **11. eicosanoid**	1. steroid **2. eicosanoid** 3. lipid 4. carbohydrate
(acquired_abnormality, co-occurs_with, injury_or_poisoning)	tail-to-head	1. congenital_abnormality **2. acquired_abnormality** 3. anatomical_abnormality 4. mental_process	1. congenital_abnormality **2. acquired_abnormality** 3. anatomical_abnormality 4. injury_or_poisoning
	head-to-tail	1. biomedical_occupation_or_discipline 2. experimental_model_of_disease 3. congenital_abnormality **10. injury_or_poisoning**	1. acquired_abnormality 2. congenital_abnormality **3. injury_or_poisoning** 4. experimental_model_of_disease
(experimental_model_of_disease, co-occurs_with, anatomical_abnormality)	tail-to-head	**1. experimental_model_of_disease** 2. anatomical_abnormality 3. injury_or_poisoning 4. professional_or_occupational_group	**1. experimental_model_of_disease** 2. anatomical_abnormality 3. injury_or_poisoning 4. sign_or_symptom
	head-to-tail	1. experimental_model_of_disease 2. disease_or_syndrome 3. neoplastic_process **6. anatomical_abnormality**	**1. anatomical_abnormality** 2. experimental_model_of_disease 3. neoplastic_process 4. disease_or_syndrome
(laboratory_procedure, assesses_effect_of, element_ion_or_isotope)	tail-to-head	**1. laboratory_procedure** 2. molecular_biology_research_technique 3. research_activity 4. therapeutic_or_preventive_procedure	**1. laboratory_procedure** 2. professional_or_occupational_group 3. professional_or_occupational_group 4. laboratory_or_test_result
	head-to-tail	1. body_substance 2. clinical_attribute 3. organism_attribute **15. element_ion_or_isotope**	1. body_substance **2. element_ion_or_isotope** 3. quantitative_concept 4. clinical_attribute
(laboratory_or_test_result, co-occurs_with, sign_or_symptom)	tail-to-head	1. professional_or_occupational_group 2. congenital_abnormality 3. disease_or_syndrome **19. laboratory_or_test_result**	**1. laboratory_or_test_result** **2.** sign_or_symptom 3. congenital_abnormality 4. acquired_abnormality
	head-to-tail	1. biomedical_occupation_or_discipline 2. occupation_or_discipline 3. intellectual_product **4. sign_or_symptom**	**1. sign_or_symptom** 2. conceptual_entity 3. biomedical_occupation_or_discipline 4. occupation_or_discipline

Table 7: Cases of the asymmetry alleviation on the UMLS dataset.