From Alignment to Entailment:
A Unified Textual Entailment Framework for Entity Alignment

Yu Zhao1 Yike Wu2 Xiangrui Cai1 Ying Zhang1∗ Haiwei Zhang3 Xiaojie Yuan3

1 College of Computer Science, TKLNDST, Nankai University, Tianjin, China
2 School of Journalism and Communication, CMRC, Nankai University, Tianjin, China
3 College of Cyber Science, TKLNDST, Nankai University, Tianjin, China
{wuyike,caixr,yingzhang,zhhaiwei,yuanxj}@nankai.edu.cn

Abstract

Entity Alignment (EA) aims to find the equivalent entities between two Knowledge Graphs (KGs). Existing methods usually encode the triples of entities as embeddings and learn to align the embeddings, which prevents the direct interaction between the original information of the cross-KG entities. Moreover, they encode the relational triples and attribute triples of an entity in heterogeneous embedding spaces, which prevents them from helping each other.

In this paper, we transform both triples into unified textual sequences, and model the EA task as a bi-directional textual entailment task between the sequences of cross-KG entities. Specifically, we feed the sequences of two entities simultaneously into a pre-trained language model (PLM) and propose two kinds of PLM-based entity aligners that model the entailment probability between sequences as the similarity between entities. Our approach captures the unified correlation pattern of two kinds of information between entities, and explicitly models the fine-grained interaction between original entity information. The experiments on five cross-lingual EA datasets show that our approach outperforms the state-of-the-art EA methods and enables the mutual enhancement of the heterogeneous information. Codes are available at https://github.com/OreOZhao/TEA.

1 Introduction

Knowledge Graphs (KGs) organize and store the facts in the real world to an effective structure, and have been applied to many knowledge-driven tasks, such as question answering (Lan et al., 2021), recommender systems (Wang et al., 2022), and information extraction (Sui et al., 2022; Zhou et al., 2021). Since the KGs are often from various domains, Entity Alignment (EA) provides fundamental techniques to find the equivalent entities in two KGs, which would complement the knowledge coverage of KGs.

∗Corresponding author.

Existing EA methods usually consist of two modules: (1) embedding module encodes entity information to entity embeddings, (2) alignment module guides the embeddings of the aligned entities to be similar (Sun et al., 2020). Moreover, they usually incorporate two kinds of heterogeneous triples as shown in Figure 1a: (1) relational triples \((h, r, t)\), represents the relation \(r\) between head entity \(h\) and tail entity \(t\), (2) attribute triples \((e, a, v)\), represents the attribute value \(v\) of the attribute \(a\) of entity \(e\).

![Figure 1](https://example.com/figure1.png)

(b) Our bi-directional entailment modeling of cross-KG entity sequences, where the sub-sequences with the same color shading share the same semantics.

Figure 1: (a) displays an example of relational and attribute information of entity "The Rolling Stones" in ZH-EN KGs.

Despite the progress of existing EA methods (Liu et al., 2020; Tang et al., 2021; Zhong et al., 2022), they are limited by the embedding-based architecture in two folds: (1) Lack of direct interaction between KGs. Existing methods usually treat EA as a representation learning task. During the encoding process, the origin triples of entities
are compressed to a continuous vector, which prevents them from directly interacting with each other. However, the origin information contains rich semantics information. Take the entity "The Rolling Stones" in Figure 1a as an example, the attribute value "Rollingstones.com" and "1962" of the Chinese KG are highly compatible with the value "The Rolling Stones" and "1962" in the English KG. The correlation between the values can directly indicate the alignment of two entities.

(2) Heterogeneous embedding spaces. Existing methods usually encode the relational triples and attribute triples in different embedding spaces due to the heterogeneity of structures and literals. This way, the alignment of relational information and of attribute information are separated and could not help each other. However, they may share the same correlation pattern. For example, the entity "The Rolling Stone" in Chinese and English KGs in Figure 1 have common neighbors (translated) and common attribute values, which could both indicate the equivalence of entities. Capturing the correlation pattern in a unified model would enable mutual enhancement between the two information.

Inspired by recent progress of pre-trained language models (PLMs) (Brown et al., 2020; Gao et al., 2021; Sun et al., 2022), we transform both two kinds of triples into textual sequences, and propose a unified Textual Entailment framework for entity Alignment TEA. We model the EA task as a bi-directional textual entailment task between the sequences of cross-KG entities as shown in Figure 1b to explicitly capture the fine-grained interaction between entity information. Specifically, we combine two sequences of entities in one sequence with cloze-style templates and feed the combined sequence into a PLM. We further propose two aligners to model the entailment probability as the pre-training tasks of PLM, i.e. Next Sentence Prediction (NSP) and Masked Language Modeling (MLM). The NSP-Aligner predicts the probability of whether one entity is next sentence of the other, while the MLM-Aligner fills in the blanks between entity sequences with mapped label words "Yes" or "No". The positive entailment probability is seen as entity similarity and is used for ranking the candidate entities. The experiments on five cross-lingual EA datasets show that TEA outperforms the state-of-the-art methods and enables the mutual enhancement of heterogeneous information.

Overall, the contributions of this paper can be summarized as follows:

- We unify the modeling of the relational triples and attribute triples in EA by transforming both into textual sequences and capturing their common correlation pattern.
- To the best of our knowledge, we are the first to transform EA to a bi-directional textual entailment task of relational and attribute information. The proposed PLM-based aligners capture the fine-grained interaction between cross-KG entities.
- Experiments on five cross-lingual EA datasets demonstrate that our approach outperforms baselines and enables the mutual enhancement of heterogeneous information.

2 Related Work

2.1 Entity Alignment

Existing EA methods usually follow an embedding-alignment architecture (Sun et al., 2020), where the entity encoder learns from the relational and attribute triples with various networks, then the alignment module guides the embeddings of the aligned entities to be similar.

There are two mainstreams of methods: TransE (Bordes et al., 2013) based methods (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Sun et al., 2018; Guo et al., 2019) for KG representation with simple implementation, and GCN (Welling and Kipf, 2016) based methods (Chen et al., 2017; Sun et al., 2017; Zhu et al., 2017; Sun et al., 2018; Guo et al., 2019) for modeling graph structures. However, the rich semantics in the origin information of cross-KG entities lack interaction through the encoding process. Our work focuses on modeling the interaction between the origin information of cross-KG entities.

For methods incorporating attribute information with relational information, they usually encode them in heterogeneous representation spaces with hybrid encoders. For example, GNNs (Sun et al., 2019; Liu et al., 2020) and RNNs (Guo et al., 2019; Zhong et al., 2022) are used for encoding relational triples to model the structures of entities, while Skip-gram (Sun et al., 2017), N-hot (Wang et al., 2018; Yang et al., 2019) and BERT (Liu et al., 2020; Zhong et al., 2022) for attribute triples for capturing literal semantics. Some methods further
aggregate the heterogeneous embeddings in separate sub-graphs (Wang et al., 2018; Yang et al., 2019; Liu et al., 2020; Tang et al., 2021). However, the heterogeneous embedding spaces hinder the EA process. Our work focuses on the unified modeling of relational and attribute information.

There have been other advancements in EA, focusing on unsupervised or self-supervised EA (Mao et al., 2021; Liu et al., 2022), incorporation of entity images (Liu et al., 2021; Lin et al., 2022), EA with dangling cases (Sun et al., 2021), which motivates our future work.

2.2 PLMs in KGs

With the prosperity of PLMs like BERT (Devlin et al., 2019), fine-tuning the PLM in downstream tasks has shown great potential in KGs. In EA, several methods have explored PLMs in learning entity embeddings (Yang et al., 2019; Tang et al., 2021; Zhong et al., 2022). However, they share the same drawbacks with methods in Section 2.1, and some methods (Yang et al., 2019; Tang et al., 2021) require extra natural language sequences such as entity descriptions which are not always available.

Recent studies (Brown et al., 2020; Gao et al., 2021; Sun et al., 2022) show that given a natural-language prompt, the PLM could achieve remarkable improvements by simulating the pre-training tasks of PLM, i.e. NSP and MLM. The prompt-based fine-tuning paradigm has been applied in many tasks in KGs, such as Named Entity Recognition (Huang et al., 2022), Entity Linking (Sun et al., 2022), Entity Typing (Ding et al., 2021). However, there is no prompt-learning study for entity-pair tasks such as EA. Our work focuses on constructing entity-pair sequences with prompts, and transforming the EA task to the NSP-style or MLM-style textual entailment task. The entailment probability is seen as entity similarity.

3 Methodology

3.1 Preliminaries

Knowledge Graph. A knowledge graph (KG) could be defined as \( \mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{V}, \mathcal{T}^r, \mathcal{T}^a\} \), where \( \mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{V} \) is the set of entities, relations, attributes and attribute values, respectively. The \( \mathcal{T}^r = \{(h, r, t) \mid h, t \in \mathcal{E}, r \in \mathcal{R}\} \) is the set of relational triples. The \( \mathcal{T}^a = \{(e, a, v) \mid e \in \mathcal{E}, a \in \mathcal{A}, v \in \mathcal{V}\} \) is the set of attribute triples.

Entity Alignment. Given the two KGs \( \mathcal{G}_1 \) and \( \mathcal{G}_2 \), the target of EA is to find a mapping between two KGs, i.e. \( \mathcal{P} = \{(e, e') \mid e \in \mathcal{G}_1, e' \in \mathcal{G}_2\} \). A set of alignment seeds \( \mathcal{P}^s \) is used as training data.

3.2 Overview

In our TEA framework, we first transform an entity as textual sequences composed of its neighbors and attribute values, and then measure the similarity between a pair of cross-KG entities via a text entailment task on their sequences. Finally, we perform the entity alignment based on similarity.

Now we elaborate on the textual entailment task. As shown in Figure 2, we first combine two sequences of cross-KG entities with a cloze-style template, and input the combined sequence into the PLM. Then, we tune the PLM with the entailment objectives to enlarge the positive entailment probability of the positive entity pairs. The entailment probability \( p(y|T(e, e')) \) is from one of the two proposed PLM-based entity aligners, NSP-Aligner or MLM-Aligner.

In practice, we find that the computationally cost is prohibitive to perform text entailment between all the entity pairs in two KGs. Therefore, besides the entailment objectives, we also tune the PLM simultaneously with the entity embedding-alignment objective, which minimizes the distance between the embeddings of the aligned entity pairs. For efficient EA inference, we first filter out the most similar candidates based on the embeddings learned from the embedding-alignment objective, and then re-rank these candidates via the entity similarity learned from the entailment objectives.

3.3 Input construction

Sequence construction. We follow previous studies (Tang et al., 2021; Zhong et al., 2022) to construct sequences with neighbors and attribute values, which contain rich semantics. For entity \( e \), the relational neighbors are \( \mathcal{N}_e = \{n|\{(e, r, n) \in \mathcal{T}^r\}\} \), and the attribute values are \( \mathcal{V}_e = \{v|\{(e, a, v) \in \mathcal{T}^a\}\} \). We sort the \( \mathcal{N}_e \) and \( \mathcal{V}_e \) in alphabetical order by relation \( r \) and attribute \( a \) to form sequences respectively. The sequences are denoted as \( S^r(e) = "e, n_1, n_2, ..., n_{|\mathcal{N}_e|}[SEP]" \), \( n_i \in \mathcal{N}_e \) and \( S^a(e) = "e, v_1, v_2, ..., v_{|\mathcal{V}_e|}[SEP]" \), \( v_i \in \mathcal{V}_e \).

Entity-pair input. Existing PLM-based EA methods usually take the weighted hidden state of [CLS] of single-entity input \( x = [CLS]S(e)[SEP] \) for entity embedding. In our work, we propose to combine the sequences of two entities together and learn from their correlation. The input could be denoted as \( T(e, e') = [CLS]S(e)[T]S(e') \), where
the S(e) and S(e') could be S^T(e) or S^a(e), and [1] could be any templates. We discuss the effect of templates in Section 4.4.

**Attention mask matrix.** As shown in Figure 2, we design an attention mask matrix M to implement the simultaneous tuning of the entailment objectives and the entity embedding-alignment objective, where the entailment mask M_0 exposes the whole entity-pair sequence to PLM and embedding masks M_1 and M_2 expose only one of the entities.

### 3.4 Training

**Training set.** In each epoch, we first construct a training set D = \{(e, e^+, e^-) \mid (e, e^+) \in \mathcal{P}^e, e^- \in \mathcal{G}_2, e^+ \neq e^-\}, where each alignment seed (e, e^+) from the training data \mathcal{P}^e has a negative counterpart e^-.

Thus the model could be trained to distinguish the positive pair (e, e^+) from the negative pair (e, e^-). We randomly select e^- from the top entities in \mathcal{G}_2 with the highest embedding cosine similarity scores with e. The embeddings for negative sample selection are obtained from the fixed PLM with single-entity input, and are consistent with the embeddings which are fine-tuned in the training phase with entity-pair input and embedding masks M_1 or M_2.

**Bi-directional training.** For learning the bi-directional correlation between entities for alignment, we tune the PLM with the bi-directional sequences, i.e. T(e, e') and T(e', e).

**Cooperated training.** For capturing the common correlation pattern of relational and attribute information, we tune the PLM with one epoch of relational input T^r(e, e') and one epoch of attribute input T^a(e, e') until convergence.

### 3.5 Embedding-Alignment Objective

The sequence T(e, e') is tokenized and put into a pre-trained language model with the attention mask, such as multilingual BERT for cross-lingual EA. We denote the obtained hidden states conditioned on the input sequence and attention mask \(M_m\) as \(H^m = \{h^m_{[CLS]}, h^m_{[SEP]}, \ldots, h^m_{[SEP]}\} = \text{PLM}(T(e, e'); M_m)\).

We obtain the embedding of entities following a standard fine-tuning paradigm. We obtain the hidden output of the PLM for the two entities \(e = W_{emb}h^1_{[CLS]}\) and \(e' = W_{emb}h^2_{[CLS]}\), where the \(W_{emb} \in \mathbb{R}^{emb \times d}\) projects the hidden size of PLM d to embedding size emb. Then we apply the pairwise margin ranking loss in the embeddings of the training set as Equation (1) to minimize the distance between the positive entity pairs and maximize the distance of negative entity pairs. The \(d(e, e')\) denotes the distance function between two entities and \(m\) is a hyper-parameter that represents the margin between the positive and negative pairs. We use \(l_2\) distance as distance function.

\[
\mathcal{L}_{mr} = \sum_{(e, e^+, e^-) \in D} \max\{0, d(e, e^+) - d(e, e^-) + m\}. \tag{1}
\]

### 3.6 Entailment Objectsives

For fully using the language modeling ability of PLMs, existing methods (Gao et al., 2021; Sun et al., 2022) propose to model the downstream task as the pre-training tasks of PLM, i.e. NSP and MLM. We propose two aligners based on the pre-training tasks of PLMs, i.e. NSP-Aligner and MLM-Aligner. Since we transform the EA task to a bi-directional text entailment task, we directly utilize NSP Head or MLM Head to represent if two entities entail each other, i.e. align to each other.
We denote the label space of entailment-style EA as $Y = \{\text{align, not-align}\}$.

**NSP-Aligner.** The origin NSP task predicts if the second sentence comes after the first sentence. For NSP-Aligner, the model predicts the probability of whether entity $e$ is after $e'$ and vice versa, to demonstrate the correlation of two entities. In this way, we can treat the entailment-style EA task as an NSP task. As shown in Equation (2), with the input of $T(e, e')$, the output of NSP head is the pre-sofmax logit $p_{\text{nsp}}$, where $n \in \{\text{next, not_next}\}$ respects to $Y$, $W_{\text{nsp}} \in \mathbb{R}^{2 \times d}$ is the weight matrix learned by NSP task, and $h_0^{[\text{CLS}]}$ is the hidden state of [CLS] with the entailment mask $M_0$.

$$p_{\text{nsp}}(y|T(e, e')) = p(y|T(e, e')) = W_{\text{nsp}}(\text{tanh}(Wh_0^{[\text{CLS}]} + b)) \tag{2}$$

**MLM-Aligner.** The origin MLM task predicts the masked token [MASK] in the sequence. For MLM-Aligner, the model learns a mapping from the label space to the set of individual words in the vocabulary, denoted as $M : Y \rightarrow V$ with label word such as "Yes" of "No". In this way, we can treat the entailment-style EA task as an MLM task.

The MLM head fills the gaps [MASK] with the label word probability as Equation (3), where $W_{\text{mlm}} \in \mathbb{R}^{V \times d}$ projects the hidden state of PLM to the vocabulary size and $h_0^{[\text{MASK}]}$ is the hidden state of [MASK] with the entailment mask $M_0$.

$$p_{\text{mlm}}(y|T(e, e')) = p([\text{MASK}] = M(y)|T(e, e')) = W_{\text{mlm}}h_0^{[\text{MASK}]} + b \tag{3}$$

**Prompt bi-directional entailment loss.** In the training phase, we train the NSP-Aligner or MLM-Aligner with two losses. The first loss is a binary cross entropy loss for prompt entailment $L_{pe}$ as shown in Equation (4) where $q(y|T(e, e')) = \text{softmax}(p(y|T(e, e')))$. We train the positive entity pair with positive label 1 and the negative pair with negative label 0. We also add the reversed $L_{pe}'$ with the input $T(e', e)$ for bi-directional modeling. The final bi-directional entailment loss is $L_{be} = L_{pe} + L_{pe}'$.

$$L_{pe} = \text{BCE}(q(y|T(e, e^+), 1) + \text{BCE}(q(y|T(e, e^-)), 0) \tag{4}$$

**Prompt bi-directional margin loss.** The second loss is the prompt margin ranking loss $L_{pmr}$ as Equation (5), where the positive probability $p^+(y|T(e, e'))$ of positive entity pairs are enlarged compared to the negative pairs. The positive probability is $p_{\text{nsp}}(y|T(e, e')) = p(n = \text{next}|T(e, e'))$ for NSP-Aligner and $p_{\text{mlm}}^+(y|T(e, e')) = p([\text{MASK}] = "Yes"|T(e, e'))$ for MLM-Aligner. We also use the bi-directional prompt margin loss as $L_{pmr} = L_{pmr} + L'_{pmr}$.

$$L_{pmr} = \sum_{(e, e^+, e^-) \in D} \max\{0, p^+(y|T(e, e^-)) - p^+(y|T(e, e^+)) + m\} \tag{5}$$

The overall objective of TEA is the sum of three losses as Equation (6).

$$L = L_{pmr} + L_{be} + L_{bnm} \tag{6}$$

### 3.7 Inference

In the inference phase, we use entity embeddings for the first few top fixed number of entity pairs, with the highest cosine similarity scores as candidate entity set $C(e)$. The candidate number $|C(e)|$ is hyper-parameter.

**Confidence-aware sample selection.** We use entity embeddings to select the candidate entity set. For each entity in $G_1$, we retrieve the top fixed number of entities from $G_2$ with the highest cosine similarity scores as candidate entity set $C(e)$. The candidate number $|C(e)|$ is hyper-parameter.

**4 Experiments**

#### 4.1 Experimental Settings

**Datasets.** To evaluate the proposed method, we conduct experiments on two widely used EA datasets: DBP15K (Sun et al., 2017) and SRPRS (Guo et al., 2019). DBP15K is the most commonly used EA dataset and consists of three cross-lingual EA subsets, which are Chinese-English (ZH-EN), Japanese-English (JA-EN), and French-English (FR-EN). SRPRS is a sparse EA dataset with much fewer triples and consists of two cross-lingual EA subsets, which are English-French (EN-FR) and English-German (EN-DE). The dataset
Table 1: Entity alignment performance on DBP15K and SRPRS. We highlight the best and the second best results of each column. The "w/o $T^a" means training the model without modeling attribute information. The TEA-NSP and TEA-MLM achieve the best or the second best in all metrics on all datasets.

Table 2: Datasets statistics for EA.
Table 3: Ablation study on DBPZH-EN. The [T] means templates. \( L_{hc} \) and \( L_{bm} \) means the prompt bi-directional entailment loss and margin loss. \( T^r \) means relational information. MLM-FT-EA is a variation of FT-EA where the entity embeddings are obtained in MLM-style.

<table>
<thead>
<tr>
<th>Template ( T(e, e') )</th>
<th>TEA-NSP</th>
<th>TEA-MLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S(e) \rightarrow [MASK] \rightarrow S(e') )</td>
<td>93.1 98.3 0.95</td>
<td>93.1 98.1 0.95</td>
</tr>
<tr>
<td>( S(e) \rightarrow [MASK] \rightarrow S(e') )</td>
<td>93.4 98.2 0.95</td>
<td>93.2 97.8 0.95</td>
</tr>
<tr>
<td>( S(e) \rightarrow [MASK] \rightarrow S(e') )</td>
<td>93.0 97.8 0.95</td>
<td>93.3 98.2 0.95</td>
</tr>
</tbody>
</table>

Table 4: Effect of templates on DBPZH-EN. For hard templates, we manually design some templates. For soft templates, we use the special token \( P_l \) following Ding et al. (2021), where \( l \) is a hyper-parameter.

4.3 Ablation Study

We conduct the ablation study as shown in Table 3. **Q1: Is the cloze-style template necessary for NSP-Aligner?** Since most prompt-learning methods use the cloze-style templates to form an MLM task rather than an NSP task, thus we remove the cloze-style template in the NSP-Aligner with TEA-NSP w/o [T], i.e. only use [SEP] token to divide the sequences of two entities. The performance declines 1.5% in Hits@1 compared to the TEA-NSP, which shows that the template could also enhance the performance of NSP-Aligner.

**Q2: Are the entailment objectives necessary?** The ablation of two entailment losses \( L_{hc} \) and \( L_{bm} \) results in a decrease of 3.8% and 0.9%, respectively. Thus two losses both enhance the re-ranking performance and the binary cross-entropy loss enhances more than the margin loss.

**Q3: Do the relational sequences and attribute sequences enhance each other?** The TEA-NSP and the TEA-NSP w/o \( T^r \) are both evaluated by methods modeling relational triples and entity names, TEA-NSP and TEA-MLM achieve the best or the second best in all metrics on all datasets. Even on DBPZH-EN where baselines fail to perform well, TEA-MLM outperforms the baselines by at most 4.4% in Hits@1 and 11% in MRR. Moreover, compared with FT-EA, the re-ranking with NSP-Aligner and MLM-Aligner brings significant improvements, at most 20.1% in Hits@1 and 15% in MRR improvements.

The TEA-NSP and TEA-MLM perform comparably on DBP15K and TEA-MLM performs better than TEA-NSP on SRPRS. The reason could be that MLM-Aligner is more competitive in the low-resource setting (Gao et al., 2021) since the SRPRS dataset has fewer triples. We will look into EA under the low-resource setting in the future.

**Comparison with group (2).** Compared with methods modeling heterogeneous triples and entity names, TEA performs the best or the second best in all metrics. The TEA-NSP outperforms the baselines by 9.3% in Hits@1 and 7% in MRR at most, and outperforms the FT-EA by 10.9% in Hits@1 and 9% in MRR at most. We could observe that BERT-INT(name) (Tang et al., 2021) performs the best or the second best in some metrics on the FR-EN, EN-FR, and EN-DE alignment. The reason could be that BERT-INT relies more on the similarity between entity names, and English shares many similar expressions with French and German. Thus BERT-INT’s performance declines on the alignment between less-alike languages.

TEA on SRPRS in group (1) and (2) are both evaluated with relational sequences. With extra attribute information, TEA in group (2) outperforms the TEA w/o \( T^a \) in group (1). It demonstrates that by modeling the common correlation pattern of the heterogeneous information with the PLM-based aligners, the extra attribute information would enhance the alignment of relational information. On the contrary, without the modeling of the common correlation, the performance of FT-EA slightly declines or stays the same on the SRPRS dataset than FT-EA w/o \( T^a \).

The TEA-NSP are comparable but slightly better than TEA-MLM in group (2). The reason could be that the interaction modeling of two aligners is similar, but NSP-Aligner is better with sentence-pair input than MLM-Aligner since NSP is designed to process sentence pairs.
attribute information. By modeling the extra relational information, the performance of evaluating with attribute information increases by 4.0% in Hits@1, which means the modeling of relational information enhances the modeling of the attribute information. Moreover, the analysis in Section 4.2 shows the reversed enhancement. They demonstrate that by modeling the common correlation of relational and attribute information in a unified manner would enable mutual enhancement.

Q4: Is the interaction of entity-pair necessary? We construct MLM-FT-EA, a variation of FT-EA, to ablate the entity-pair interaction with reservation of the prompt learning. Inspired by recent progress in sentence embedding (Jiang et al., 2022), we use a cloze-style template: This sentence of “S(e)” means [MASK]. To obtain entity embeddings with MLM-FT-EA. The performance of MLM-FT-EA is similar to FT-EA. It shows that the entity-pair interaction is the most important component in TEA rather than the prompt-learning paradigm.

4.4 Effect of Templates

In this section, we study the effect of templates in TEA. As stated by previous studies (Gao et al., 2021; Tam et al., 2021), the templates have impacts on the performance of prompt-learning oriented tasks. We design both hard templates and soft templates on DBPZHI-EN dataset. The hard templates are manually designed, while the soft templates have a varying number of learnable special prompt tokens following Ding et al. (2021). As shown in Table 4, the templates could affect the performance of EA considerably. For hard templates, the I know that improves the performance the most. For soft templates, the TEA-NSP needs fewer special tokens while the TEA-MLM needs more.

4.5 Effect of Re-ranking Parameters

Figure 3 shows the hyper-parameter analysis of the re-ranking process of TEA. The sample number is the number of entities in G1 to be re-ranked by the PLM-based aligners. With a higher threshold, more samples are re-ranked and the performance of EA is better. When threshold $\delta = 0.9$, the re-ranking samples are 37% less than re-ranking all the samples ($\delta = 1.0$) but the performance is similar and the re-ranking time cost are highly reduced.

The candidate number is the number of entities in $G_2$ that are most likely to be the ground truth. With more candidates, the performance is better. The reason could that the ground truth entity is more likely to be in the candidate set when the candidate set is larger. Moreover, even with only 16 candidates, the performance of TEA in Hits@1 exceeds the FT-EA by 7.6%.

4.6 Case Study

We conduct a case study as shown in Table 5, trying to find the aligned entity of Singapour (FR). The entity ranking conducted by embeddings shows the best-aligned entity is Thailand (EN). However, by re-ranking the candidates with PLM-based aligners, the fine-grained interaction between entities is explicitly modeled. As shown in the visualization, the Singapour (FR)-Singapore (EN) pair has more at-
tentative sub-sequences (darker diagonal short lines) while the unaligned pair Singapour (FR)-Thailand (EN) have not. Moreover, the aligned entity is ranked first place by the PLM-based aligner.

5 Conclusion

To address the limitations of the existing EA method, the lack of interaction and heterogeneous embedding spaces, we propose a unified textual entailment framework for entity alignment called TEA. We transform the origin relational triples and attribute triples of an entity into textual sequences and model the EA task as a bi-directional textual entailment task between the sequences of cross-KG entities. We propose two kinds of PLM-based aligners to capture the fine-grained correlation between entities with two kinds of sequences in a unified manner. The entailment probability is used for measuring entity similarity and ranking the entity candidates. Experiment results on five cross-lingual datasets show that TEA outperforms existing EA methods and enables the mutual enhancement between the heterogeneous information.

Limitations

Despite that TEA achieves some gains for EA, TEA still has the following limitations:

First, TEA has a higher computation cost than the embedding-based EA methods in the re-ranking phase, since TEA process entity-pair input for modeling the interaction between them. For reducing time costs, we adopt the confidence-aware re-ranking strategy to reduce the number of re-ranking samples and candidates. However, the inference time cost is still higher than the embedding-based methods. In addition, the candidate selection may be limited in some corner cases if the ground truth entity is not ranked in the top $|C|$ similar entities calculated by entity embeddings. We will further explore efficient approaches which could cover the corner cases.

Second, the alignment of relational information of TEA requires the entity names to construct sequences. However, the entity names are not always available in some EA datasets, such as the Wikidata KG in OpenEA Benchmark (Sun et al., 2020). In that case, TEA can use the attribute sequences without entity names for entity alignment. Though TEA w/o $T^*$ can achieve competitive performance as shown in Table 3, it still limits the application of TEA. We will further explore PLM-based approaches to align the relational information without the requirement of entity names.

Acknowledgements

This research is supported by the Natural Science Foundation of Tianjin, China (No. 22JCJQJC00150, 22JCQNJC01580), the National Natural Science Foundation of China (No. 62272250, U1936206, 62002178), Tianjin Research Innovation Project for Postgraduate Students (No. 2022SKYZ232), and the Fundamental Research Funds for the Central Universities (No. 63231149, 63232114).

References


Xuhui Sui, Ying Zhang, Kehui Song, Baohang Zhou, Guoqing Zhao, Xin Wei, and Xiaojie Yuan. 2022. Improving zero-shot entity linking candidate generation with ultra-fine entity type information. In COLING, pages 2429–2437.


Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. In IJCAI.


ACL 2023 Responsible NLP Checklist

A  For every submission:

✔ A1. Did you describe the limitations of your work?
   *Section Limitations after the Section 5 Conclusion.*

☐ A2. Did you discuss any potential risks of your work?
   *Not applicable. Left blank.*

✔ A3. Do the abstract and introduction summarize the paper’s main claims?
   *Section Abstract and Section 1 Introduction.*

✘ A4. Have you used AI writing assistants when working on this paper?
   *Left blank.*

B  ✔ Did you use or create scientific artifacts?
   *Section 3 Methodology and Section 4 Experiments.*

✔ B1. Did you cite the creators of artifacts you used?
   *Section 4 Experiments.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
   *Not applicable. Left blank.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
   *Not applicable. Left blank.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
   *Not applicable. Left blank.*

✔ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   *Section 4 Experiments.*

✔ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
   *Section 4 Experiments.*

C  ✔ Did you run computational experiments?
   *Section 4 Experiments.*

✔ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
   *Section 4 Experiments.*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*
C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4 Experiments and GitHub codes.

C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 Experiments.

C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 Experiments.

D □ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants’ demographic (e.g., country of residence)?
No response.

D3. Did you discuss whether and how consent was obtained from people whose data you’re using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.