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

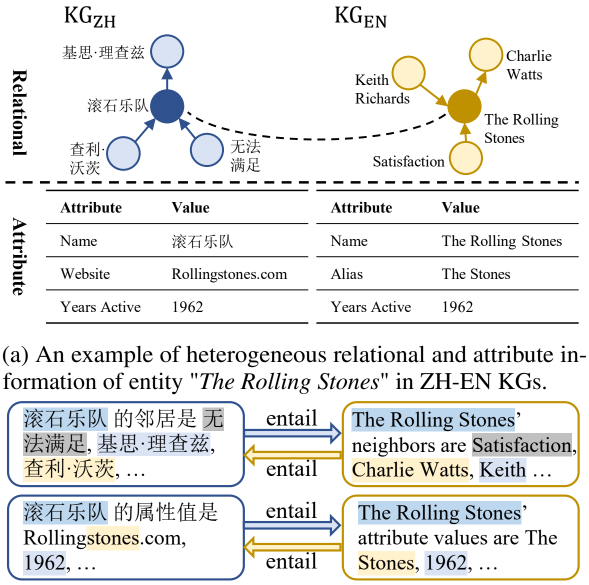
Yu Zhao[1] **Yike Wu**[2] **Xiangrui Cai**[1] **Ying Zhang**[1]*
Haiwei Zhang[3] **Xiaojie Yuan**[1]
[1] College of Computer Science, TKLNDST, Nankai University, Tianjin, China
[2] School of Journalism and Communication, CMRC, Nankai University, Tianjin, China
[3] College of Cyber Science, TKLNDST, Nankai University, Tianjin, China
zhaoyu@dbis.nankai.edu.cn
{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.



(a) An example of heterogeneous relational and attribute information of entity "*The Rolling Stones*" in ZH-EN KGs.

(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 entities. (b) displays our bi-directional entailment modeling for EA.

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 r, (2) attribute triples (e, a, v), represents the attribute value v of the attribute a of entity e.

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

[*] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 8795–8806
July 9-14, 2023 ©2023 Association for Computational Linguistics

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 **T**extual **E**ntailment framework for entity **A**lignment **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. $\mathcal{P} = \{(e, e') | e \in \mathcal{G}_1, e' \in \mathcal{G}_2\}$. 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 $S^r(e) = "e, n_1, n_2, ..., n_{|\mathcal{N}_e|}\texttt{[SEP]}", n_i \in \mathcal{N}_e$ and $S^a(e) = "e, v_1, v_2, ..., v_{|\mathcal{V}_e|}\texttt{[SEP]}", v_i \in \mathcal{V}_e$.

Entity-pair input. Existing PLM-based EA methods usually take the weighted hidden state of `[CLS]` of single-entity input $x = \texttt{[CLS]}S(e)\texttt{[SEP]}$ 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 $T(e, e') = \texttt{[CLS]}S(e)\texttt{[T]}S(e')$, where

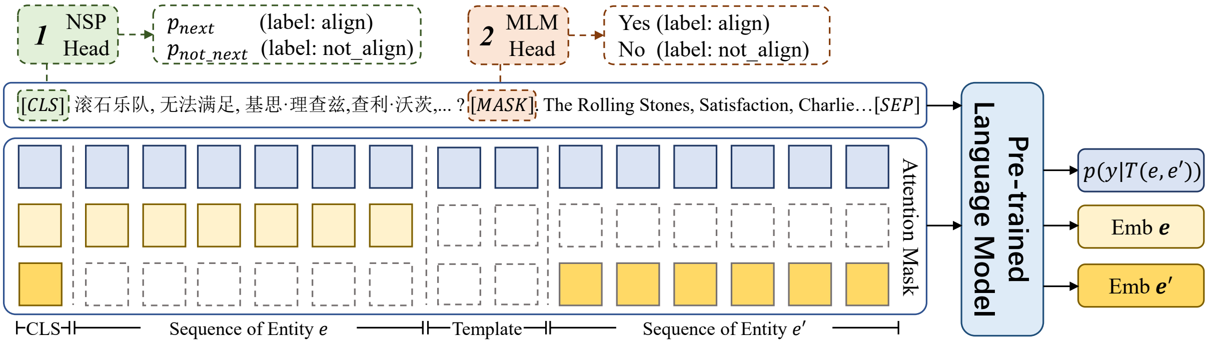

1 NSP Head
p_{next} (label: align)
p_{not_next} (label: not_align)
2 MLM Head
Yes (label: align)
No (label: not_align)
[CLS] 滚石乐队, 无法满足, 基思·理查兹,查利·沃茨,... ? [MASK] The Rolling Stones, Satisfaction, Charlie…[SEP]
Attention Mask
Pre-trained Language Model
$p(y|T(e,e'))$
Emb e
Emb e'
CLS
Sequence of Entity e
Template
Sequence of Entity e'

Figure 2: The architecture of TEA, textual entailment framework for entity alignment. The input of PLM is the entity-pair sequence of (e, e') and the attention mask we design for tuning PLM with both entailment and embedding-alignment objectives. The output of PLM is the probability of entailment $p(y|T(e, e'))$ and the embeddings of two entities $\mathbf{e}$ and $\mathbf{e}'$. The probability is from either NSP Head of NSP-Aligner or MLM Head of MLM-Aligner.

the $S(e)$ and $S(e')$ could be $S^r(e)$ or $S^a(e)$, and `[T]` 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 $\mathcal{D} = \{(e, e^+, e^-)|(e, e^+) \in \mathcal{P}^s, e^- \in \mathcal{G}_2, e^+ \neq e^-\}$, where each alignment seed (e, e^+) from the training data $\mathcal{P}^s$ 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 $H^m = \{\mathbf{h}^m_{\texttt{[CLS]}}, \mathbf{h}^m_1, ..., \mathbf{h}^m_l, \mathbf{h}^m_{\texttt{[SEP]}}\} = \texttt{PLM}(T(e, e'); M_m)$.

We obtain the embedding of entities following a standard fine-tuning paradigm. We obtain the hidden output of the PLM for the two entities $\mathbf{e} = \mathbf{W}_{\text{emb}}\mathbf{h}^1_{\texttt{[CLS]}}$ and $\mathbf{e}' = \mathbf{W}_{\text{emb}}\mathbf{h}^2_{\texttt{[CLS]}}$, where the $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{emb \times d}$ 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 $d(\mathbf{e}, \mathbf{e}')$ 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.

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

3.6 Entailment Objectives

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 $\mathcal{Y} = \{\texttt{align, not\_align}\}$.

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-softmax logit p_{nsp}, where $n \in \{\texttt{next, not\_next}\}$ respects to $\mathcal{Y}$, $\mathbf{W}_{\text{nsp}} \in \mathbb{R}^{2\times d}$ is the weight matrix learned by NSP task, and $\mathbf{h}^0_{\texttt{[CLS]}}$ is the hidden state of `[CLS]` with the entailment mask M_0.

$$\begin{aligned} p_{nsp}(y|T(e,e')) &= p(n|T(e,e')) \\ &= \mathbf{W}_{\text{nsp}}(tanh(\mathbf{W}\mathbf{h}^0_{\texttt{[CLS]}} + \mathbf{b})) \end{aligned} \tag{2}$$

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 $\mathcal{M} : \mathcal{Y} \rightarrow \mathcal{V}$ 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 $\mathbf{W}_{\text{mlm}} \in \mathbb{R}^{V\times d}$ projects the hidden state of PLM to the vocabulary size and $\mathbf{h}^0_{\texttt{[MASK]}}$ is the hidden state of `[MASK]` with the entailment mask M_0.

$$\begin{aligned} p_{mlm}(y|T(e,e')) &= p(\texttt{[MASK]} = \mathcal{M}(y)|T(e,e')) \\ &= \mathbf{W}_{\text{mlm}}\mathbf{h}^0_{\texttt{[MASK]}} + \mathbf{b} \end{aligned} \tag{3}$$

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 $\mathcal{L}_{pe}$ as shown in Equation (4) where $q(y|T(e,e')) = \texttt{softmax}(p(y|T(e,e')))$. We train the positive entity pair with positive label 1 and the negative pair with negative label 0. We also add the reversed $\mathcal{L}'_{pe}$ with the input $T(e', e)$ for bi-directional modeling. The final bi-directional entailment loss is $\mathcal{L}_{be} = \mathcal{L}_{pe} + \mathcal{L}'_{pe}$.

$$\mathcal{L}_{pe} = \texttt{BCE}(q(y|T(e,e^+),1) + \texttt{BCE}(q(y|T(e,e^-)),0) \tag{4}$$

Prompt bi-directional margin loss. The second loss is the prompt margin ranking loss $\mathcal{L}_{pmr}$ 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 $p^+_{nsp}(y|T(e,e')) = p(n = \texttt{next}|T(e,e'))$ for NSP-Aligner and $p^+_{mlm}(y|T(e,e')) = p(\texttt{[MASK]} = \text{"}\textit{Yes}\text{"}|T(e,e'))$ for MLM-Aligner. We also use the bi-directional prompt margin loss as $\mathcal{L}_{bm} = \mathcal{L}_{pmr} + \mathcal{L}'_{pmr}$.

$$\begin{aligned} \mathcal{L}_{pmr} = \sum_{(e,e^+,e^-)\in\mathcal{D}} & max\{0, p^+(y|T(e,e^-)) \\ & - p^+(y|T(e,e^+)) + m\} \end{aligned} \tag{5}$$

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

$$\mathcal{L} = \mathcal{L}_{mr} + \mathcal{L}_{be} + \mathcal{L}_{bm} \tag{6}$$

3.7 Inference

In the inference phase, we use entity embeddings for the first ranking. Then we use the PLM-based aligner, NSP-Aligner or MLM-Aligner, for re-ranking the hard samples with the candidates selected by the entity embeddings.

Candidate entity selection. We use entity embeddings to select the candidate entity set. For each entity in $\mathcal{G}_1$, we retrieve the top fixed number of entities from $\mathcal{G}_2$ with the highest cosine similarity scores as candidate entity set $\mathcal{C}(e)$. The candidate number $|\mathcal{C}(e)|$ is hyper-parameter.

Confidence-aware sample selection. We use the highest similarity score between e and entities in $\mathcal{C}(e)$ as the embedding confidence score for sample e, denoted as $c(e) = max\{cos(\mathbf{e}, \mathbf{e}')|e' \in \mathcal{C}(e)\}$. We assume that the test samples with lower confidence scores are harder samples for embeddings to obtain accurate results. Then we re-rank the samples with lower confidence than a fixed threshold $c(e) < \delta$, with the positive probability $p^+(y|T(e,e'))$ of PLM-based aligner. The samples with higher confidence use the similarity of embeddings as final alignment results. The threshold δ 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

Method	$\mathrm{DBP}_{\mathrm{ZH-EN}}$			$\mathrm{DBP}_{\mathrm{JA-EN}}$			$\mathrm{DBP}_{\mathrm{FR-EN}}$			$\mathrm{SRPRS}_{\mathrm{EN-FR}}$			$\mathrm{SRPRS}_{\mathrm{EN-DE}}$		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
Methods modeling relational triples and entity names															
RDGCN	69.7	84.2	0.75	76.3	89.7	0.81	87.3	95.0	0.90	67.2	76.7	0.71	77.9	88.6	0.82
HGCN	70.8	84.0	0.76	75.8	88.9	0.81	88.8	95.9	0.91	67.0	77.0	0.71	76.3	86.3	0.80
CEA(Emb)	71.9	85.4	0.77	78.5	90.5	0.83	92.8	98.1	0.95	93.3	97.4	0.95	94.5	98.0	0.96
CEA	78.7	-	-	86.3	-	-	**97.2**	-	-	96.2	-	-	97.1	-	-
FT-EA w/o $\mathcal{T}^a$	67.5	91.0	0.76	68.9	90.8	0.77	95.8	99.3	**0.97**	96.7	98.8	0.97	97.0	**99.1**	**0.98**
TEA-NSP w/o $\mathcal{T}^a$	**81.5**	**95.3**	**0.87**	**89.0**	**96.7**	**0.92**	**96.8**	**99.5**	**0.98**	**97.3**	**99.4**	**0.98**	**97.2**	**99.6**	**0.98**
TEA-MLM w/o $\mathcal{T}^a$	**83.1**	**95.7**	**0.88**	**88.3**	**96.6**	**0.91**	**96.8**	**99.4**	**0.98**	**98.1**	**99.5**	**0.99**	**98.3**	**99.6**	**0.99**
Methods modeling relational triples, attribute triples, and entity names															
AttrGNN	79.6	92.9	0.85	78.3	92.1	0.83	91.9	97.8	0.91	-	-	-	-	-	-
BERT-INT(name)	81.4	83.5	0.82	80.6	83.5	0.82	**98.7**	99.2	**0.99**	**97.1**	97.5	**0.97**	**98.6**	98.8	**0.99**
SDEA	87.0	96.6	0.91	84.8	95.2	0.89	96.9	99.5	0.98	96.6	98.6	**0.97**	96.8	98.9	**0.98**
FT-EA	85.4	95.7	0.89	83.2	93.4	0.87	95.7	99.0	0.97	96.4	98.9	**0.97**	97.0	99.1	**0.98**
TEA-NSP	**94.1**	**98.3**	**0.96**	**94.1**	**97.9**	**0.96**	**97.9**	**99.7**	**0.99**	**98.5**	**99.6**	**0.99**	**98.7**	**99.6**	**0.99**
TEA-MLM	**93.5**	**98.2**	**0.95**	**93.9**	**97.8**	**0.95**	**98.7**	**99.6**	**0.99**	**98.5**	**99.6**	**0.99**	**98.7**	**99.7**	**0.99**

Table 1: Entity alignment performance on DBP15K and SRPRS. We highlight the **best** and the **second best** results of each column. The "w/o $\mathcal{T}^a$" 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.

statistics of DBP15K and SRPRS are listed in Table 2. Consistent with previous studies, we randomly choose 30% of the samples for training and 70% for testing.

Evaluation metrics. We use Hits@K (K=1,10), which is the accuracy in top K predictions, and Mean Reciprocal Rank (MRR), which is the average reciprocal ranking of ground-truth entity, as evaluation metrics. The higher Hits@K and higher MRR indicate better performance.

Implementation details. We implement our approach with Pytorch and Transformers (Wolf et al., 2020). We use BERT (Devlin et al., 2019) as the PLM for cross-lingual EA following Liu et al. (2020); Tang et al. (2021); Zhong et al. (2022). The information for evaluating TEA is one of the relational and attribute information which performs higher Hits@1 in the validation set, i.e. attribute for DBP15K and relational for SRPRS. The training is early stopped after 3 epochs of no improvements of Hits@1 in the validation set. We conduct the experiments in Ubuntu 18.04.5 with a single NVIDIA A6000 GPU with 48GB of RAM.

Dataset		$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{T}^r\rvert$	$\lvert\mathcal{T}^a\rvert$	$\lvert\mathcal{P}\rvert$
$\mathrm{DBP}_{\mathrm{ZH-EN}}$	ZH	19,388	1,701	7,780	70,414	379,684	15,000
	EN	19,572	1,323	6,933	95,142	567,755	
$\mathrm{DBP}_{\mathrm{JA-EN}}$	JA	19,814	1,299	5,681	77,214	354,619	15,000
	EN	19,780	1,153	5,850	93,484	497,230	
$\mathrm{DBP}_{\mathrm{FR-EN}}$	FR	19,661	903	4,431	105,998	528,665	15,000
	EN	19,993	1,208	6,161	115,722	576,543	
$\mathrm{SRPRS}_{\mathrm{EN-FR}}$	EN	15,000	221	274	36,508	70,750	15,000
	FR	15,000	177	393	33,532	56,344	
$\mathrm{SRPRS}_{\mathrm{EN-DE}}$	EN	15,000	222	275	38,363	62,715	15,000
	DE	15,000	120	185	37,377	142,506	

Table 2: Datasets statistics for EA.

Baselines. To comprehensively evaluate our method TEA, the baselines are grouped into two categories according to the input information. Since we construct the sequences with entity names, we mainly compare TEA with the method that also models entity names. (1) The methods modeling relational triples and entity names: RDGCN (Wu et al., 2019a), HGCN (Wu et al., 2019b), CEA (Zeng et al., 2020). (2) The methods modeling relational triples, attribute triples, and entity names: AttrGNN (Liu et al., 2020), BERT-INT (Tang et al., 2021), SDEA (Zhong et al., 2022). For BERT-INT which uses entity descriptions, we replace the descriptions with entity names for a fair comparison following Zhong et al. (2022).

We construct a baseline FT-EA, which learns and inferences with the entity embeddings for alignment results. FT-EA could be seen as TEA w/o textual entailment objectives and re-ranking. We report the results of TEA with two PLM-based aligners, TEA-NSP and TEA-MLM. We also ablated the attribute sequence (w/o $\mathcal{T}^a$) to compare with the baselines of group (1).

4.2 Comparison with Baselines.

We compare our method with the baselines and the results are presented in Table 1.

Comparison with group (1). Compared with

	$\text{DBP}_{\text{ZH-EN}}$		
	Hits@1	Hits@10	MRR
TEA-NSP	94.1	98.3	0.96
TEA-NSP w/o [T]	92.6	97.7	0.95
TEA-NSP w/o $\mathcal{L}_{be}$	90.3	97.4	0.93
TEA-NSP w/o $\mathcal{L}_{bm}$	93.2	98.0	0.95
TEA-NSP w/o $\mathcal{T}^r$	90.1	97.1	0.93
MLM-FT-EA	85.2	95.2	0.89

Table 3: Ablation study on $\text{DBP}_{\text{ZH-EN}}$. The [T] means templates. $\mathcal{L}_{be}$ and $\mathcal{L}_{bm}$ means the prompt bidirectional entailment loss and margin loss. $\mathcal{T}^r$ means relational information. MLM-FT-EA is a variation of FT-EA where the entity embeddings are obtained in MLM-style.

Template $T(e, e')$	TEA-NSP			TEA-MLM		
	H@1	H@10	MRR	H@1	H@10	MRR
Hard templates						
$S(e)$? [MASK]. $S(e')$	93.3	98.1	0.95	93.2	98.1	0.95
$S(e)$? [MASK]. I know that $S(e')$	93.6	97.8	0.95	93.4	97.8	0.95
$S(e)$? [MASK]. I think that $S(e')$	92.3	97.4	0.94	93.2	97.8	0.95
Soft templates						
$S(e)$ [MASK][P$_0$]...[P$_l$] $S(e')$, l=1	94.1	98.3	0.96	92.8	97.8	0.95
$S(e)$ [MASK][P$_0$]...[P$_l$] $S(e')$, l=2	93.4	97.8	0.95	93.2	97.9	0.95
$S(e)$ [MASK][P$_0$]...[P$_l$] $S(e')$, l=3	92.8	97.8	0.95	93.3	98.2	0.95
$S(e)$ [MASK][P$_0$]...[P$_l$] $S(e')$, l=4	92.5	97.8	0.95	93.5	98.2	0.95

Table 4: Effect of templates on $\text{DBP}_{\text{ZH-EN}}$. 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.

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 $\text{DBP}_{\text{ZH-EN}}$ 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 $\mathcal{T}^a$ 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 $\mathcal{T}^a$.

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.

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 $\mathcal{L}_{be}$ and $\mathcal{L}_{bm}$ 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 $\mathcal{T}^r$ are both evaluated by

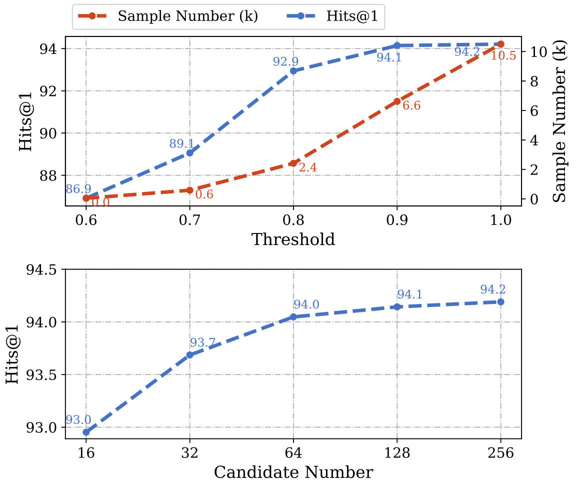


Figure 3: Re-ranking parameter analysis conducted by TEA-NSP on $\mathrm{DBP}_{\mathrm{ZH-EN}}$.

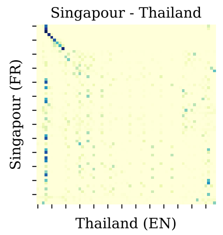
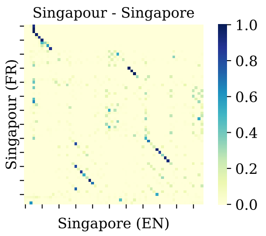
Before Re-ranking	**After Re-ranking**
Visualization of Attention Weights	
	
Rankings of Entity Candidates	
Thailand , Malaysia, Hong Kong, Khmer language, Khuang Aphaiwong, Cambodia, Patna, Singapore , ...	Singapore , Indonesian language, Javanese language, West Sumatra, Malaysia, Thailand , Jakarta Cambodia, Patna, ...

Table 5: The attention weights of entity pair with highest alignment probability and the entity rankings before and after re-ranking.

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 $\mathrm{DBP}_{\mathrm{ZH-EN}}$ 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 $\mathcal{G}_1$ 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 $\mathcal{G}_2$ 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 attentive 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 $|\mathcal{C}|$ 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 $\mathcal{T}^r$ 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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *NeurIPS*, 26.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *NeurIPS*, 33:1877–1901.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *ACL*, pages 3816–3830.

Lingbing Guo, Zequn Sun, and Wei Hu. 2019. Learning to exploit long-term relational dependencies in knowledge graphs. In *ICML*, pages 2505–2514. PMLR.

Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. 2022. Copner: Contrastive learning with prompt guiding for few-shot named entity recognition. In *COLING*, pages 2515–2527.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. 2022. Promptbert: Improving bert sentence embeddings with prompts. *arXiv preprint arXiv:2201.04337*.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. *arXiv preprint arXiv:2105.11644*.

Zhenxi Lin, Ziheng Zhang, Meng Wang, Yinghui Shi, Xian Wu, and Yefeng Zheng. 2022. Multi-modal contrastive representation learning for entity alignment. In *COLING*, pages 2572–2584.

Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021. Visual pivoting for (unsupervised) entity alignment. In *AAAI*, pages 4257–4266.

Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2022. Selfkg: self-supervised entity alignment in knowledge graphs. In *Proceedings of the ACM Web Conference 2022*, pages 860–870.

Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. In *EMNLP*, pages 6355–6364.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. From alignment to assignment: Frustratingly simple unsupervised entity alignment. In *EMNLP*, pages 2843–2853.

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.

Yi Sun, Yu Zheng, Chao Hao, and Hangping Qiu. 2022. Nsp-bert: A prompt-based few-shot learner through an original pre-training task——next sentence prediction. In *COLING*, pages 3233–3250.

Zequn Sun, Muhao Chen, and Wei Hu. 2021. Knowing the no-match: Entity alignment with dangling cases. In *ACL*, pages 3582–3593.

Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644. Springer.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, volume 18, pages 4396–4402.

Zequn Sun, Jiacheng Huang, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. Transedge: Translating relation-contextualized embeddings for knowledge graphs. In *International Semantic Web Conference*, pages 612–629. Springer.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment*, 13(12).

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. 2021. Improving and simplifying pattern exploiting training. In *EMNLP*, pages 4980–4991.

Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2021. Bert-int: a bert-based interaction model for knowledge graph alignment. In *IJCAI*, pages 3174–3180.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *ACM SIGKDD*, pages 1929–1937.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357.

Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric 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 *EMNLP*, pages 38–45.

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

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019b. Jointly learning entity and relation representations for entity alignment. In *EMNLP-IJCNLP*, pages 240–249.

Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. 2019. Aligning cross-lingual entities with multi-aspect information. In *EMNLP-IJCNLP*, pages 4431–4441.

Weixin Zeng, Xiang Zhao, Jiuyang Tang, and Xuemin Lin. 2020. Collective entity alignment via adaptive features. In *ICDE*, pages 1870–1873. IEEE.

Ziyue Zhong, Meihui Zhang, Ju Fan, and Chenxiao Dou. 2022. Semantics driven embedding learning for effective entity alignment. In *ICDE*, pages 2127–2140. IEEE.

Baohang Zhou, Xiangrui Cai, Ying Zhang, and Xiaojie Yuan. 2021. An end-to-end progressive multi-task learning framework for medical named entity recognition and normalization. In *ACL*, pages 6214–6224.

Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Iterative entity alignment via knowledge embeddings. 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.