Align-then-Enhance: Multilingual Entailment Graph Enhancement with Soft Predicate Alignment

Yuting Wu[1], **Yutong Hu**[2,3], **Yansong Feng**[2,3]*, **Tianyi Li**[4]
Mark Steedman[4], **Dongyan Zhao**[2,3]
[1]School of Software Engineering, Beijing Jiaotong University, China
[2]Wangxuan Institute of Computer Technology, Peking University, China
[3]The MOE Key Laboratory of Computational Linguistics, Peking University, China
[4]School of Informatics, University of Edinburgh, U.K.
ytwu1@bjtu.edu.cn, {huyutong,fengyansong,zhaody}@pku.edu.cn
tianyi.li@ed.ac.uk, steedman@inf.ed.ac.uk

Abstract

Entailment graphs (EGs) with predicates as nodes and entailment relations as edges are typically incomplete, while EGs in different languages are often complementary to each other. In this paper, we propose a new task, multilingual entailment graph enhancement, which aims to utilize the entailment information from one EG to enhance another EG in a different language. The ultimate goal is to obtain an enhanced EG containing richer and more accurate entailment information. We present an align-then-enhance framework (ATE) to achieve accurate multilingual entailment graph enhancement, which first exploits a cross-graph guided interaction mechanism to automatically discover potential equivalent predicates between different EGs and then constructs more accurate enhanced entailment graphs based on soft predicate alignments. Extensive experiments show that ATE achieves better and more robust predicate alignment results between different EGs, and the enhanced entailment graphs generated by ATE outperform the original graphs for entailment detection[1].

1 Introduction

Predicate entailment detection is the task to determine if the meaning of one predicate is entailed (can be inferred) from the other predicate, which benefits many core natural language processing tasks such as question answering and semantic parsing. Usually, a question like "Did Lionel Messi *appear at* the FIFA World Cup Stadium Gelsenkirchen in 2006" might be answered by a sentence that does not directly correspond to the question, but is an expression like "Lionel Messi *made his debut at* the FIFA World Cup Stadium Gelsenkirchen in 2006", since the predicate *"make one's debut at"* entails predicate *"appear at"*. To bridge such semantic gap between queries and answers, recent approaches (Berant et al., 2011, 2015; Hosseini et al., 2018, 2019; Hosseini, 2021) have looked into learning typed Entailment Graphs (EGs) with predicates as nodes and entailment relations as edges. These methods first extract predicate-argument (entity) relation triples from large text corpora, and construct typed entailment graphs based on the Distributional Inclusion Hypothesis (Dagan et al., 1999; Geffet and Dagan, 2005; Herbelot and Ganesalingam, 2013; Kartsaklis and Sadrzadeh, 2016). Predicates are then grouped into typed entailment subgraphs based on the types of entity arguments they take. Such an EG is in an effective and machine-readable form to organize the context-independent entailment relations between predicates and can facilitate reasoning without extra context or resources, which can be regarded as a special kind of Knowledge Graph (KG) for natural language understanding. Figure 1 shows excerpts from two typed EGs in different languages with arguments of types <*person, location*>.

However, EGs frequently suffer from incompleteness, and EGs in different languages are often complementary to each other. This makes a compelling case for developing a method that can integrate entailment information from EGs in different languages and construct an enhanced EG containing richer and more accurate entailment information. For instance, to answer the aforementioned question "Did Lionel Messi *appear at* the FIFA World Cup Stadium Gelsenkirchen in 2006", given

*Corresponding author.
[1]Code and data available at https://github.com/StephanieWyt/Align-then-Enhance.

Findings of the Association for Computational Linguistics: ACL 2023, pages 881–894
July 9-14, 2023

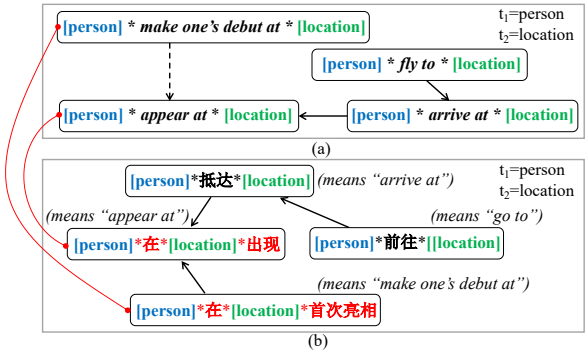

Figure 1: (a) and (b) are English and Chinese entailment graph examples with arguments of types *person* and *location*, respectively. Each red line connects a pair of equivalent predicates between two EGs, and the dashed arrow in (a) indicates a new entailment edge enhanced from the premise between two red predicates in (b).

the sentence "Lionel Messi *made his debut at* the FIFA World Cup Stadium Gelsenkirchen in 2006", we queried the popular English EG published by Hosseini et al. (2018) and found no entailment edge where *"make one's debut at"* entails *"appear at"*. However, as shown in Figure 1 (b), in the Chinese EG constructed by Li et al. (2022), we find an entailment edge where the predicate "在·X·首次亮相" entails "在·X·出现" . Significantly, "在·X·首次亮相" and *"make one's debut at"* as well as "在·X·出现" and *"appear at"* are two pairs of equivalent predicates. If such equivalent predicates between Chinese and English EGs could be aligned, we can use the entailment information in one EG to enhance another. For example, according to the Chinese entailment edge where "在·X·首次亮相" entails "在·X·出现", we can add the equivalent entailment edge where *"make one's debut at"* entails *"appear at"* to the English EG. The enhanced EG fuses the entailment information from different EGs, further boosting the entailment detection performance of the original graph.

Recently, a few efforts have been made to improve the quality of an EG by integrating entailment information from another EG. Weber and Steedman (2019) have tried to align the English and German EGs by learning the predicate representations towards alignment through a link prediction model and showed that the stronger English EG can aid in German entailment detection. Whereas Weber and Steedman (2019) only proves that an EG in a higher resource language can improve the quality of an EG in a lower resource language, Li et al. (2022) further demonstrates that the cross-lingual complementarity between different EGs can be used in both directions by ensembling the predictions from the two graphs. However, they did not really realize the alignment of EGs. In order to achieve an ensemble, their model needs the parallel Chinese translations of the English questions to query the Chinese and English EGs separately.

In this paper, we propose a new task, *Multilingual Entailment Graph Enhancement* (MEGE), which aims to automatically align EGs in different languages and utilize the entailment information from one EG to enhance the other. We emphasize that the enhancement should be effective in both directions, which is demonstrated and discussed in Section 6.2.

For the multilingual entailment graph enhancement task, we present an *align-then-enhance* framework, ATE, which first automatically discovers equivalent predicates between EGs in different languages and then constructs more accurate enhanced EGs based on soft predicate alignments. In order to achieve accurate predicate alignment, an effective approach is to learn better predicate representations toward alignment. However, since EGs usually suffer from severe sparsity issues, it is difficult to directly embed EG structures to learn effective predicate representations. For example, 79% of nodes in the popular English EG released by Hosseini et al. (2018) have degrees no more than 2. To tackle this issue, we introduce rich context information of predicates extracted from the large open-domain encyclopedia KG Wikidata (Vrandecic and Krötzsch, 2014) into original EG to assist in learning better predicate representations, constructing a *predicate-centric graph*. Simultaneously, we also build the *entity-centric graph*, aiming to improve the predicate representations by utilizing the information of entities closely associated to the predicates. And we propose a cross-graph guided interaction (CGI) mechanism to encourage sufficient interaction between the predicate-centric graph and the entity-centric graph and learn better predicate representations for alignment.

To our best knowledge, there is no standard dataset to directly evaluate the predicate alignment performance for entailment graphs. We thus build a new alignment evaluation dataset *EGAlign*. Experiments on EGAlign indicate that our model achieves the state-of-the-art performance on predicate alignment between different EGs. The key technical contributions of this paper are as follows:

- We propose a new task, multilingual entailment graph enhancement (MEGE), to improve the quality of an EG with another EG in a different language and construct an enhanced EG which can better support entailment detection. We present an align-then-enhance framework for this task.

- We design a cross-graph guided interaction mechanism to overcome the sparsity of EGs, which encourages the information interaction between the enriched predicate-centric graph and entity-centric graph and learns better predicate representations towards alignment.

- We build a new EG predicate alignment evaluation dataset, and our model achieves the state-of-the-art performance on it. We further evaluate the enhanced EGs on two benchmark datasets for entailment detection, and demonstrate that the enhanced entailment graphs are superior to the original graphs.

2 Related Work

Most previous works on entailment graphs only focused on constructing an entailment graph in a single language, and usually ignore the complementarity between different EGs. Similar to the idea of entailment graph enhancement, (Lewis and Steedman, 2013b) do not construct an enhanced entailment graph but learned clusters of semantically equivalent English and French predicates based on their named-entity arguments. They create predicate representations and align the predicates by the cosine similarity between representations. (Lewis and Steedman, 2013a) solve the problem of paraphrasing in a multilingual context by creating aligned paraphrase clusters. They take the Wikipedia articles describing the same topic as parallel texts, and use the Wikipedia inter-language links between named entities to align predicates in different languages. The study on creating paraphrase clusters lays a foundation for the construction of entailment graphs. Berant et al. (2015) first propose typed predicate entailment graphs but can not be scalable to a large amount of data. To tackle this problem, Hosseini et al. (2018) present a scalable method that learns globally consistent similarity scores for entailment graph construction.

Based on the typed entailment graphs, Weber and Steedman (2019) are the first to try to align an English entailment graph with a German entailment graph. However, as discussed in Section 1, the serious sparsity issues of EGs hinder their alignment performance and they only perform alignment in one direction. Most recently, Li et al. (2022) demonstrate the cross-lingual complementarity between an English EG and a Chinese EG in both directions. However, instead of really aligning the two EGs, they ensemble the predictions from the two graphs, which needs the parallel translations between English and Chinese questions for querying the English and Chinese EGs separately.

3 Problem Formulation

Let P be the set of all typed predicates and T be the set of types, $\mathcal{V}(t_1, t_2)$ denotes the set of typed predicates p with unordered argument types t_1 and t_2, where $p \in P$ and $t_1, t_2 \in T$. The argument types of each predicate are determined by the types of entities that instantiate the argument slots. Formally, we represent a typed EG as $\mathcal{G}(t_1, t_2) =< \mathcal{V}(t_1, t_2), \mathcal{E}(t_1, t_2) >$, where $\mathcal{V}(t_1, t_2)$ is the set of typed predicate nodes and $\mathcal{E}(t_1, t_2)$ denotes the set of weighted edges. We represent the edges as the sparse score matrix $W(t_1, t_2) \in [0, 1]^{|\mathcal{V}(t_1,t_2)|\times|\mathcal{V}(t_1,t_2)|}$, containing the entailment scores between predicates of types t_1 and t_2.

Without loss of generality, we consider the entailment graph enhancement task between two typed EGs, $\mathcal{G}_1(t_1, t_2) =< \mathcal{V}_1(t_1, t_2), \mathcal{E}_1(t_1, t_2) >$ and $\mathcal{G}_2(t_1, t_2) =< \mathcal{V}_2(t_1, t_2), \mathcal{E}_2(t_1, t_2) >$. The goal of this task is to utilize the entailment information (i.e., the entailment scores between typed predicates) from one EG to enhance the other. In this paper, we achieve this in two steps: 1) Given a set of pre-aligned typed predicate pairs $\mathbb{L} = \{(p_{i_1}, p_{i_2}) | p_{i_1} \in \mathcal{V}_1(t_1, t_2), p_{i_2} \in \mathcal{V}_2(t_1, t_2)\}$ between $\mathcal{G}_1(t_1, t_2)$ and $\mathcal{G}_2(t_1, t_2)$, our approach first finds more aligned typed predicates as much as possible based on the existing ones; 2) With these predicate alignments in place, we utilize $\mathcal{G}_2(t_1, t_2)$ to enhance $\mathcal{G}_1(t_1, t_2)$, and construct an enhanced entailment graph $\hat{\mathcal{G}}^{2\to1}(t_1, t_2) =< \hat{\mathcal{V}}(t_1, t_2), \hat{\mathcal{E}}(t_1, t_2) >$ with updated entailment score matrix $\hat{W}(t_1, t_2)$. Similarly, we can also obtain the enhanced entailment graph $\hat{\mathcal{G}}^{1\to2}(t_1, t_2)$ generated by using $\mathcal{G}_1(t_1, t_2)$ to enhance $\mathcal{G}_2(t_1, t_2)$.

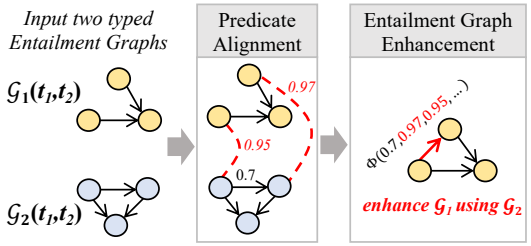

Figure 2: ATE architecture. 0.97 and 0.95 are learned alignment scores of predicate pairs between two EGs, and 0.7 is the original entailment score of that edge.

4 Our Approach: ATE

To obtain an enhanced EG, we propose a model, ATE. As depicted in Figure 2, ATE takes two typed EGs as input, and follows a two-stage pipeline: predicate alignment and entailment graph enhancement, and finally output an enhanced EG with richer and more accurate entailment information.

4.1 Predicate Alignment

In this stage, we aim to discover as many aligned predicate pairs as possible between $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$. As discussed in Section 1, EGs often suffer from sparsity issues, which makes it difficult to learn good node (predicate) representations. To tackle this problem, we use context information of predicates extracted from Wikidata to enrich original EGs and construct denser predicate-centric graphs. We also introduce entity-centric graphs with entities as nodes and predicates as edges, which provide rich information for entities closely associated with the predicates. We propose a cross-graph guided interaction mechanism to encourage sufficient interaction between predicate-centric and entity-centric graphs, learning better predicate representations for alignment.

4.1.1 Entity/Predicate-centric Graph Construction

Let E_1 and E_2 be the entity instances of the argument slots of typed predicates in $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$. We take entities as nodes and predicates as edges to construct the ***entity-centric graphs*** $G_1^e = (E_1^e, P_1^e, T_1^e)$ and $G_2^e = (E_2^e, P_2^e, T_2^e)$ for $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$, respectively, where E_1^e and E_2^e are the entity sets and P_1^e and P_2^e are the predicate sets and $T_1^e \subset E_1^e \times P_1^e \times E_1^e$ and $T_2^e \subset E_2^e \times P_2^e \times E_2^e$ are the binary relation triples. Besides the structural information contained in the predicate-entity relation triples, we also introduce the neighborhood information of entities in Wikidata to further enrich the entity-centric graphs. Therefore, the node sets $E_1^e = E_1 \cup N_1$ and $E_2^e = E_2 \cup N_2$, where N_1 and N_2 are the one-hop neighbors extracted from Wikidata[2]; the predicate sets $P_1^e = \mathcal{V}_1(t_1,t_2) \cup P_1^{wiki}$ and $P_2^e = \mathcal{V}_2(t_1,t_2) \cup P_2^{wiki}$, where P_1^{wiki} and P_2^{wiki} are the Wikidata predicates associated with N_1 and N_2, respectively.

We also add the newly introduced Wikidata predicates P_1^{wiki} and P_2^{wiki} into $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$ respectively to enhance the connectivity of two EGs and introduce richer contextual information of typed predicates of $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$, obtaining the ***predicate-centric graphs*** $G_1^p = (\mathcal{V}_1^p, \mathcal{E}_1^p)$ and $G_2^p = (\mathcal{V}_2^p, \mathcal{E}_2^p)$, where $\mathcal{V}_1^p = P_1^e = \mathcal{V}_1(t_1,t_2) \cup P_1^{wiki}$ and $\mathcal{V}_2^p = P_2^e = \mathcal{V}_2(t_1,t_2) \cup P_2^{wiki}$ are the node sets, and $\mathcal{E}_1^p$ and $\mathcal{E}_2^p$ are the edge sets. Besides the existing entailment edges, if two predicates share the same head or tail entities in the entity-centric graphs, we will create an edge connecting the two predicate nodes v_i^p and v_j^p, and weight the edge with weight s_{ij} according to how likely the two predicates share similar heads or tails in entity-centric graphs:

$$s_{ij} = \frac{|H_i \cap H_j|}{|H_i \cup H_j|} + \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \tag{1}$$

where H_i and T_i are the sets of head and tail entities for predicates p_i in entity-centric graphs. Considering the original entailment scores on the entailment edges together, the final weight ϵ_{ij}^p between predicate nodes v_i^p and v_j^p is computed as:

$$\epsilon_{ij}^p = \begin{cases} s_{ij}, & v_i^p \text{ or } v_j^p \in P_1^{wiki} \cup P_2^{wiki} \\ w_{ij} + s_{ij}, & v_i^p, v_j^p \in \mathcal{V}_1(t_1,t_2) \cup \mathcal{V}_2(t_1,t_2) \end{cases} \tag{2}$$

where w_{ij} is the entailment score between predicate p_i and p_j in original EGs. Note that we also modify the entailment scores of gold edges, which allows us to additionally incorporate richer features of typed predicates from the introduced world knowledge.

In order to facilitate the implementation of our model, we put G_1^e and G_2^e together as the ***final entity-centric graph*** $G^e = (E^e, P^e, T^e)$, where $E^e = E_1^e \cup E_2^e$ and $P^e = P_1^e \cup P_2^e$ and $T^e = T_1^e \cup T_2^e$. Similarly, we put G_1^p and G_2^p together as the ***final predicate-centric graph*** $G^p = (\mathcal{V}^p, \mathcal{E}^p)$, where $\mathcal{V}^p = \mathcal{V}_1^p \cup \mathcal{V}_2^p$ and $\mathcal{E}^p = \mathcal{E}_1^p \cup \mathcal{E}_2^p$.

[2]We simply link entities in E_1 and E_2 to Wikidata entities through the exact match of entity names. However, not all entities can be linked to Wikidata, so we introduce one-hop neighbors of entities that can be linked to Wikidata.

4.1.2 Cross-graph Guided Interaction

With the entity-centric graph G^e and the predicate-centric graph G^p in place, we propose a cross-graph guided interaction (CGI) mechanism to encourage interactions between the two graphs to obtain better predicate representations for alignment, which utilizes a graph attention mechanism (GAT) guided by counterpart graph to learn the node representations of G^e and G^p iteratively. Each cross-graph guided interaction consists of two layers, the predicate attention layer and the entity attention layer. By stacking multiple interactions, we can achieve more mutual improvements on both graphs. We further apply two Graph Convolutional Networks (GCNs) with highway gates on G^e and G^p respectively to model their structural information. The final predicate representations will be used to determine whether two predicates should be aligned.

Predicate Attention Layer. Let $\mathbf{X}^p \in \mathbb{R}^{|\mathcal{V}^p| \times d}$ denote the input node representation matrix of G^p. Different from the vanilla GAT (Veličković et al., 2018), each node integrates its neighbor node information under the guidance of G^e. Specifically, the predicate attention score regarding each neighbor node is calculated with the entity node features $\hat{\mathbf{X}}^e$ (computed by Eq. 9) produced by the entity attention layer from the previous interaction module:

$$\tilde{\boldsymbol{x}}_i^p = ReLU(\sum_{j \in N_i^p} \alpha_{ij}^p \boldsymbol{x}_j^p), \tag{3}$$

$$\alpha_{ij}^p = \frac{exp(\eta(\epsilon_{ij}^p a^p[\boldsymbol{z}_i \| \boldsymbol{z}_j]))}{\sum_{k \in N_i^p} exp(\eta(\epsilon_{ik}^p a^p[\boldsymbol{z}_i \| \boldsymbol{z}_k]))}, \tag{4}$$

where $\tilde{\boldsymbol{x}}_i^p$ is the output representation of predicate node v_i^p; $\boldsymbol{x}_j^p$ is the representation of v_j^p produced by the previous predicate attention layer; N_i^p indicates the set of neighbor indices of v_i^p; $a^p[\cdot]$ is a fully connected layer; η is the Leaky ReLU; $\boldsymbol{z}_i$ is the approximate predicate representation for predicate p_i , which is computed as:

$$\boldsymbol{z}_i = [\frac{\sum_{k \in H_i} \hat{\boldsymbol{x}}_k^e}{|H_i|} \| \frac{\sum_{l \in T_i} \hat{\boldsymbol{x}}_l^e}{|T_i|}], \tag{5}$$

where $\hat{\boldsymbol{x}}_k^e$ and $\hat{\boldsymbol{x}}_l^e$ are the output representations of the k-th head entity and l-th tail entity of predicate p_i from previous entity attention layer. Note that we use Glove (Pennington et al., 2014) word embeddings of predicate names to initialize these predicate nodes, which are useful features for predicate alignment. To retain this useful information, we integrate the initial features $\{\boldsymbol{x}_i^{p\_init}\}$ with the output of predicate attention layer as follows:

$$\hat{\boldsymbol{x}}_i^p = \varphi_l^p * \tilde{\boldsymbol{x}}_i^p + \boldsymbol{x}_i^{p\_init}, \tag{6}$$

where $\hat{\boldsymbol{x}}_i^p$ denotes the final output predicate representation of the interaction module for predicate node v_i^p; φ_l^p is a weighting hyper-parameter for the l-th attention layer. We show the effectiveness of this skip connection design in Section 6.1.

Entity Attention Layer. Similar to the predicate attention layer, we apply GAT on the entity-centric graph guided by the predicate-centric graph. Specifically, let $\mathbf{X}^e \in \mathbb{R}^{|E^e| \times d}$ be the input node representation matrix of G^e. The representation $\tilde{\mathbf{x}}_q^e$ of entity e_q in G^e can be computed as:

$$\tilde{\boldsymbol{x}}_q^e = ReLU(\sum_{t \in N_q^e} \alpha_{qt}^e \boldsymbol{x}_t^e), \tag{7}$$

$$\alpha_{qt}^e = \frac{exp(\eta(a^e[\hat{\boldsymbol{x}}_{qt}^p]))}{\sum_{k \in N_q^e} exp(\eta(a^e[\hat{\boldsymbol{x}}_{qk}^p]))}, \tag{8}$$

where $\hat{\boldsymbol{x}}_{qt}^p$ is the representation for the predicate between entity e_q and e_t obtained from G^p. We also initialize the entity node representations with entity names, and the final output representation $\hat{\boldsymbol{x}}_q^e$ of the interaction module for entity e_q are the weighted sum of the initial entity representations and the output of entity attention layer:

$$\hat{\boldsymbol{x}}_q^e = \varphi_c^e * \tilde{\boldsymbol{x}}_q^e + \boldsymbol{x}_q^{e\_init}, \tag{9}$$

where φ_c^e is a weighting hyper-parameter for the c-th entity attention layer.

Graph Structure Embedding. After multiple rounds of interaction between G^p and G^e, we can obtain enhanced predicate and entity representations. Following previous practice (Rahimi et al., 2018; Wu et al., 2019a), we respectively feed the two graphs into two different two-layer GCNs (Kipf and Welling, 2017) with highway gates (Srivastava et al., 2015) to incorporate evidence from their neighboring structures.

Training. With the final predicate representations $\bar{X}^p$ output by Highway-GCNs, predicate alignment can be performed by simply measuring the distance between two predicates:

$$d(p_1, p_2) = 1 - cos(\bar{\boldsymbol{x}}_1^p, \bar{\boldsymbol{x}}_2^p), \tag{10}$$

For training, we expect the distance between aligned predicate pairs to be as close as possible,

and the distance between negative predicate pairs to be as far as possible. We use the following margin-based scoring function as the training objective for predicate alignment.

$$L^p = \sum_{(p,q)\in\mathbb{L}^p} \sum_{(p',q')\in\mathbb{L}'^p} \max\{0, d(p,q) - d(p',q') + \gamma^p\}, \tag{11}$$

where $\gamma^p > 0$ is a margin hyper-parameter; $\mathbb{L}^p$ indicates the pre-aligned predicate pairs for training; $\mathbb{L}'^p$ is the set of negative instances generated through nearest neighbor sampling (Kotnis and Nastase, 2017).

Similarly, with the final entity representations $\bar{X}^e$, we can also calculate the training loss for entity alignment like Eq. 11, and learn the alignment-oriented entity representations. Predicate alignment and entity alignment can enhance each other in our model, and ultimately achieve more accurate alignment results.

4.2 Entailment Graph Enhancement

After obtaining the final alignment-oriented representation of each predicate in $\mathcal{G}_1(t_1,t_2)$ and $\mathcal{G}_2(t_1,t_2)$, we perform *soft predicate alignment* between two EGs by computing an alignment score $\pi(p^1,p^2)$ for each predicate pair (p^1,p^2) where $p^1 \in \mathcal{V}_1(t_1,t_2)$ and $p^2 \in \mathcal{V}_2(t_1,t_2)$. Specifically, we calculate the cosine similarity of the representations of p^1 and p^2. Next, we will perform EG enhancement according to these alignment scores.

As discussed in Section 1 and 3, EG enhancement can be performed in two directions. Here, we take the enhancement process of $\hat{\mathcal{G}}^{2\to1}(t_1,t_2)$ as an example. Given (p_i^1,p_j^1) as a predicate pair in $\mathcal{G}_1(t_1,t_2)$ and w_{ij} as the original entailment score between them, we aim to find the predicate pair (p_x^2,p_y^2) in $\mathcal{G}_2(t_1,t_2)$, which is aligned with (p_i^1,p_j^1), and enhance w_{ij} based on the entailment score w_{xy} between p_x^2 and p_y^2. Specifically, for (p_i^1,p_j^1), we collect the top k similar predicates of p_i^1 and p_j^1 in $\mathcal{G}_2(t_1,t_2)$ as $TopK^2(p_i^1)$ and $TopK^2(p_j^1)$, respectively. Then, we can get a set of candidate aligned predicate pairs from $\mathcal{G}_2(t_1,t_2)$, namely $C^2(p_i^1,p_j^1) = \{(p_x^2,p_y^2)|p_x^2 \in TopK^2(p_i^1), p_y^2 \in TopK^2(p_j^1)\}$.

We combine the entailment scores of all candidate predicate pairs in $C^2(p_i^1,p_j^1)$, according to their alignment probability, which is computed as:

$$\bar{w}_{ij}^2 = \frac{\sum_{(x,y)\in C^2(i,j)} AVG(\pi(i,x),\pi(j,y)) * w_{xy}}{|C^2(i,j)|}. \tag{12}$$

EGAlign	#Ent.	#Pre.	#Tri.	Alignments #Ent.pair	Alignments #Pre.pair
ZH	25,983	3,020	199,762	3,028	823
EN	13,306	4,864	126,105		

Table 1: Summary of the EGAlign dataset. "#Ent.", "#Pre." and "#Tri." indicate the number of entities, predicates and relation triples in each language, respectively.

Finally, the new enhanced entailment score $\hat{w}_{ij}^{2\to1}$ between p_i^1 and p_j^1 is updated as:

$$\hat{w}_{ij}^{2\to1} = AVG(\rho^{2\to1} * \bar{w}_{ij}, w_{ij}). \tag{13}$$

where $\rho^{2\to1}$ is a hyperparameter tuned on the development set, and AVG$(\cdot)$ denotes average pooling.

5 Experimental Setup

5.1 Predicate Alignment Evaluation

Datasets. Since there is no publicly available benchmark dataset to directly evaluate the predicate alignment performance for EGs, we construct a new alignment evaluation dataset, *EGAlign*, based on the popular English entailment graph (EG_{en}) released by Hosseini et al. (2018) and the Chinese entailment graph (EG_{zh}) constructed by Li et al. (2022). The labeling of aligned predicate pairs between two EGs is relatively labor-intensive, so we only manually aligned a set of equivalent predicates with argument types *person* and *location*. Thus, we extract the typed entailment subgraphs $EG_{en}^{p,l}$ and $EG_{zh}^{p,l}$ with arguments of types <*person,location*> from EG_{en} and EG_{zh} respectively. We annotated 5784 Chinese-English predicate pairs with three annotators per pair and reached an average inter-annotator agreement of 0.83 considering the same annotation of a pair as an agreement. In addition, we link the argument entities from $EG_{en}^{p,l}$ and $EG_{zh}^{p,l}$ respectively to English and Chinese versions of Wikidata, and obtain a set of aligned entity pairs through the inter-language links from entities of English version of Wikidata to those in Chinese. Table 1 shows the statistics of EGAlign, and we provide the annotation details for EGAlign in Appendix A. Following previous works (Wu et al., 2019b; Zhu et al., 2021), we use 30% of the pre-aligned predicate pairs and entity pairs as training data and 70% for testing.

Metrics. We use Mean Average Precision at K (MAP@K) as the evaluation metric for predicate alignment, and report the results of both directions of alignment. As discussed in Section 1,

Models	EN→ ZH			ZH → EN		
	MAP@1	MAP@10	MAP@50	MAP@1	MAP@10	MAP@50
RDGCN	2.96	4.94	5.40	2.98	3.83	4.22
BootEA	25.90	27.32	27.73	25.06	26.16	26.47
HGCN	25.14	27.45	27.81	23.08	24.45	24.84
RNM	27.31	28.94	29.70	25.56	26.45	28.02
Glove-sim	32.36	43.03	44.02	37.92	47.59	48.33
BERT-sim	31.26	41.39	42.18	33.95	41.38	42.24
CGI	**38.45**	**53.16**	**51.44**	**40.68**	**54.04**	**54.84**
w/o Wikidata	26.73	46.13	47.08	28.45	47.34	48.38
w/o GCN	34.33	47.80	48.85	37.27	49.63	50.47
w/o interaction	35.40	49.71	50.81	38.43	51.83	52.77
w/o relSkip	28.89	42.36	43.25	34.37	45.34	46.27
w/o entSkip	14.11	16.95	17.82	12.16	15.74	16.62
w/o bothSkip	11.05	13.85	14.67	10.18	12.63	13.40

Table 2: Performance on predicate alignment. For our CGI and its variants, we report the average results of 5 runs.

currently, there is no public and complete implementation of predicate alignment for EGs, we thus compare our CGI with 4 state-of-the-art models , *BootEA* (Sun et al., 2018), *RDGCN* (Wu et al., 2019a), *HGCN-JE/JR* (Wu et al., 2019b) and *RNM* (Zhu et al., 2021), for the knowledge graph relation alignment task, which is similar in spirit to our task. We also implement two baselines **Glove-sim** and **BERT-sim**, which directly take the Glove word embeddings (Pennington et al., 2014) and pre-trained BERT (Devlin et al., 2019) representations at [CLS] tokens of predicate names as the representations of predicates and perform predicate alignment by calculating the distance between them.

5.2 Entailment Detection

Datasets. To evaluate the quality of enhanced EGs and explore whether it can better support entailment detection than original graphs, we use the popular entailment detection datasets Levy/Holt (Levy and Dagan, 2016; Holt, 2019) and Berant (Berant et al., 2011). Each example in these two datasets contains a premise and a hypothesis (a pair of relation triples with the same arguments), and the entailment detection task is to judge whether the premise entails the hypothesis.

After doing EG enhancement according to Section 4.2, we obtain the enhanced EG $Enhan^{p,l}_{zh\to en}$ generated by using $EG^{p,l}_{zh}$ to enhance $EG^{p,l}_{en}$. By enhancing in the opposite direction, we can get the enhanced $Enhan^{p,l}_{en\to zh}$ as well. For fair comparison, we extract subsets with the types of arguments *person* and *location*, the portions of which are 6.3% (6107 examples) and 7.1% (2756 examples) in Levy/Holt and Berant datasets respectively. We also translated these subsets into Chinese to evaluate the performance of $Enhan^{p,l}_{en\to zh}$ and $EG^{p,l}_{zh}$. Following Hosseini et al. (2018), we split Levy/Holt dataset into development (30%) and test (70%) sets. And we evaluate our model on the test set of Levy/Holt and the whole Berant dataset. We also compare with the BERT-based baselines for etailment detection. We provide more details of the evaluation process and the construction of the BERT-based baselines in Appendix B.

Metrics. Following Hosseini et al. (2018), we evaluate our methods on LevyHolt and Berant with the area under curves (AUC) of Precision-Recall Curves (PRC). Hosseini et al. (2018) mentioned that AUC for precisions in the range [0; 0.5) should not be taken into account, since model performs as random guess and is not applicable to down-stream applications. We thus report the AUC of PRC with precision range in [0.5, 1]. More details of our configuration please refer to Appendix C.

6 Experimental Results

6.1 Predicate Alignment

From Table 2, CGI substantially outperforms all baselines across all metrics and alignment directions. The four KG relation alignment models all deliver inferior performance on EGAlign. This might be because these models approximate predicate representations via entity representations which are inferior to ours in achieving predicate alignment. We observe that Glove-sim and BERT-sim outperform other baselines, showing the importance of the semantics of predicate names.

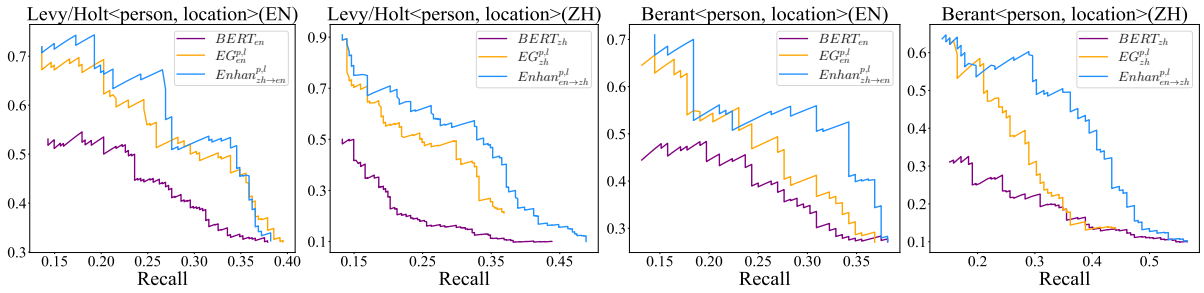

Figure 3: PRC of different methods on the <*person,location*> subsets of Levy/Holt and Berant datasets.

Ablation Study. Without introducing the Wikidata entities and predicates into our entity/predicate-centric graphs (CGI w/o Wikidata), there is a significant performance drop of 11.98 % on average regarding MAP@1. This shows the benefits of the additional Wikidata information for learning enhanced predicate representations. Removing the GCNs from our model leads to an average drop of 3.77% on MAP@1, showing the importance of the GCNs in capturing the structural information of the predicate/entity-centric graphs. When removing the cross-graph interaction, we see an average drop of 2.65% on MAP@1. This confirms the effectiveness of our cross-graph guided interaction mechanism. To explore the impact of the skip connection design that retains the useful predicate and entity name semantics contained in the initial node embeddings, we implement three model variants CGI w/o relSkip, CGI w/o entSkip and CGI w/o bothSkip, which respectively deletes the skip connections from predicate attention layer, entity attention layer and both attention layers. We observe that the three variants all deliver much worse results than CGI, which demonstrates the effectiveness of keeping the initial semantics of predicate names using skip connections.

6.2 Entailment Detection

From Table 3, we observe that, on both English and Chinese datasets, $EG^{p,l}_{en}$ and $EG^{p,l}_{zh}$, as well as $Enhan^{p,l}_{zh\rightarrow en}$ and $Enhan^{p,l}_{en\rightarrow zh}$, all greatly outperform BERT. This shows that entailment graphs seem to be more beneficial for entailment detection than BERT. Compared with $EG^{p,l}_{en}$ and $EG^{p,l}_{zh}$, the enhanced graphs $Enhan^{p,l}_{zh\rightarrow en}$ and $Enhan^{p,l}_{en\rightarrow zh}$ show obvious advantages, with average improvements of 3.7% and 6.7% on Levy/Holt and Berant subsets. Besides, as shown in Figure 3, in the moderate precision range, $Enhan^{p,l}_{zh\rightarrow en}$ and $Enhan^{p,l}_{en\rightarrow zh}$ can achieve higher recall than original graphs $EG^{p,l}_{en}$ and $EG^{p,l}_{zh}$ across all datasets, and both significantly outperform BERT. These results demonstrate the significance of performing entailment graph enhancement as well as the effectiveness of our enhancement model.

Methods	Levy/Holt	Berant
BERT_{en}	14.3	-
$\text{EG}^{p,l}_{en}$	23.0	17.6
$\text{Enhan}^{p,l}_{zh\rightarrow en}$	**24.8**	**23.2**
BERT_{zh}	8.1	-
$\text{EG}^{p,l}_{zh}$	19.9	16.0
$\text{Enhan}^{p,l}_{en\rightarrow zh}$	**26.0**	**23.3**

Table 3: AUC (%) of PRC on the <*person,location*> subsets of LevyHolt and Berant. The precision of BERT on Berant is less than 0.5 in all thresholds, thus we do not report its result here.

In addition, we note that $EG^{p,l}_{en}$ outperforms $EG^{p,l}_{zh}$ by 3.1% and 1.6% on two datasets respectively, indicating that $EG^{p,l}_{en}$ is more complete in entailment information. Hence after enhancement, comparing to the original EGs, $Enhan^{p,l}_{en\rightarrow zh}$ achieves the improvements of 6.1% and 7.3% on Levy/Holt and Berant, which are both greater than $Enhan^{p,l}_{zh\rightarrow en}$'s improvement of 1.8% and 5.6%. The results further demonstrate that the enhancement between two EGs are effective in both directions, and using stronger graphs to enhance the relatively poor ones will bring greater improvements.

Error Analysis. We conduct a case study to further analyze the circumstances where the predictions of the original EG are correct while the enhanced EGs have made wrong predictions. We randomly sampled and analyzed 100 examples that were pre-

Error Types	Examples
Inaccurate entailment scores in EG_{zh} (52%)	EN:return.to→ stay.in *with a entailment score 0.055* ZH: 重返("return to")→ 待在("stay in") *with a entailment score 0* **(inaccurate)**
Near-synonym Alignment (4%)	EN: **deliver.speech.at** → appear.at ZH: 在·**X**·主持**("preside at")**→ 在·X·演出("perform at")
Antonym Alignment (5%)	EN: **sentence.in**→appear.in.court.in ZH: 在·**X**·释放**("release in")** → 处于("be in")
Hyponym Alignment (8%)	EN: deliver.speech.at→ **appear.at** ZH: 在·X·主持("preside at") → 在·**X**·演出**("perform at")**
Unrelated Alignment (31%)	EN: **battle.in** →keep.troops.in ZH: 在·**X**·逛**("stroll around")**→ 远离("far away from")

Table 4: Major error types of $Enhan^{p,l}_{zh\rightarrow en}$.

dicted accurately by EG_{zh} but wrongly predicted by $Enhan^{p,l}_{zh\rightarrow en}$ from Levy/Holt. As shown in Table 4, the error type *Inaccurate Entailment Scores in* EG_{zh} indicates that the aligned predicate pairs predicted by ATE have inaccurate entailment scores in EG_{zh}, which negatively affect the original entailment scores in EG_{en} and lead to inaccurate updated entailment scores for $Enhan^{p,l}_{zh\rightarrow en}$. This error type accounts for more than half of the total (52%), which shows that the quality of EGs has a significant impact on the achievement of accurate EG enhancement. With EGs of higher quality, our method could generate better enhanced EG. The remaining errors are basically due to inaccurate predicate alignment between two EGs. Specifically, in 4% of the cases, the English predicates are incorrectly aligned to their synonyms in Chinese, in 5% to their antonyms, in 8% to their hyponyms (namely the Chinese predicates entail the English predicates), and in 31% to the unrelated predicates. These results inspire us to improve the quality of predicate alignment by further distinguishing synonyms, antonyms, and hyponyms in the future.

7 Conclusion

We present a new task, multilingual entailment graph enhancement, aiming to enhance the quality of one EG with another graph in a different language. We design an align-then-enhance method for this task, which utilizes a cross-graph guided interaction mechanism to tackle the sparsity issues of EGs and achieves EG enhancement based on soft predicate alignment between different EGs. Experiments show that our align-then-enhance framework can effectively mine equivalent predicates in other EGs through sufficient cross-graph interaction and better achieve accurate enhancement. We build a new dataset EGAlign to evaluate the predicate alignment performance of our model, and the results show that our model achieves the best performance. Furthermore, we show that the enhanced EGs outperform the original graph as well as BERT on entailment detection.

Limitations

There are two main limitations of our work: (1) Our approach requires a set of previously aligned predicate pairs as training data to achieve predicate alignment between different KGs, which limits the generalization ability of our method. In our experiments, since we manually aligned a set of equivalent predicates with arguments of types *person* and *location* between the English and Chinese EGs, we can only perform predicate alignment and entailment graph enhancement between the <*person,location*> subgraphs of two EGs. We will explore the semi-supervised or unsupervised predicate alignment method between different EGs in our future work. (2) Our current enhancement strategy introduced in Section 4.2 is straightforward. It might not be robust enough when dealing with entailment graphs of poor quality. We will explore more adaptive EG enhancement methods in the future.

Ethics Statement

In this paper, we construct a new EG alignment evaluation dataset based on two publicly available EGs, and manually annotated a set of equivalent predicates with argument types *person* and location. Annotators are offered a competitive pay of ¥60 per hour, which is more than double the local minimum wage. This remuneration applies to both the annotation stage and the discussion stage, ensuring that annotators are compensated for their time and effort. Annotators are required to familiarize themselves with the ACM Code of Ethics and Professional Conduct and promptly report any instances that violate the code. Inappropriate cases that breach the code are promptly eliminated from the selected documents. The resulting annotations, based on the consensus of three annotators, provide a respectable approximation of the gold labels. Note that they may not represent the absolute ground truth due to natural error rates. Users who wish to utilize the dataset should be mindful of its limitations. We are not responsible for problems en-

countered in subsequent model training processes utilizing our data.

Acknowledgements

This work is supported by the Talent Fund of Beijing Jiaotong University (No. 2023XKRC032), NSFC (62161160339), ERC Advanced Fellowship GA 742137 SEMANTAX, a Mozilla PhD scholarship at Informatics Graduate School and the University of Edinburgh Huawei Laboratory. We would like to thank the anonymous reviewers for their helpful comments and suggestions. For any correspondence, please contact Yansong Feng.

References

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Mach. Learn.*, 34(1-3):43–69.

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.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Aurélie Herbelot and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Sofia, Bulgaria. Association for Computational Linguistics.

Xavier Holt. 2019. Probabilistic models of relational implication.

Mohammad Javad Hosseini. 2021. Unsupervised learning of relational entailment graphs from text.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R. Holt, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2018. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy. Association for Computational Linguistics.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2016. Distributional inclusion hypothesis for tensor-based composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2849–2860, Osaka, Japan. The COLING 2016 Organizing Committee.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Bhushan Kotnis and Vivi Nastase. 2017. Analysis of the impact of negative sampling on link prediction in knowledge graphs. *CoRR*, abs/1708.06816.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2013a. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Mike Lewis and Mark Steedman. 2013b. Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 681–692, Seattle, Washington, USA. Association for Computational Linguistics.

Tianyi Li, Sabine Weber, Mohammad Javad Hosseini, Liane Guillou, and Mark Steedman. 2022. Cross-lingual inference with A chinese entailment graph. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1214–1233. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. 2018. Semi-supervised user geolocation via graph convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2009–2019, Melbourne, Australia. Association for Computational Linguistics.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4396–4402.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Sabine Weber and Mark Steedman. 2019. Construction and alignment of multilingual entailment graphs for semantic inference. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 77–79, Florence, Italy. Association for Computational Linguistics.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019a. Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5278–5284. ijcai.org.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2019b. Jointly learning entity and relation representations for entity alignment. 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 240–249, Hong Kong, China. Association for Computational Linguistics.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. 2020. Neighborhood matching network for entity alignment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6477–6487, Online. Association for Computational Linguistics.

Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021. Relation-aware neighborhood matching model for entity alignment. In *AAAI*.

A Annotation Details for EGAlign Dataset

In this section, we will introduce the specific alignment rules when constructing the EGAlign datasets.

For entity alignment, we link the argument entities from $EG^{p,l}_{en}$ and $EG^{p,l}_{zh}$ respectively to English and Chinese versions of Wikidata, and obtain a set of aligned entity pairs through the inter-language links from entities of English version of Wikidata to those in Chinese. For predicates alignment, human annotators are asked to grade scores (1-3) to the sampled relation pairs according to criterion as follow: **Score 1:** *Seldom Align.* Two predicates can not replace each other in any context. **Score 2:** *Sometimes Align.* Two predicates have similar usage and interpretation in some contexts (e.g., polysemy). **Score 3:** *Always Align.* Two predicates have the same usage and semantics in any context. The predicate pairs with average score greater than 1.5 are selected as predicate alignment seeds. We recruited the annotators from our school, and they are college students who are proficient in Chinese and English. Before starting annotation, annotators were informed what we will use this dataset for and the data collection protocol was approved by an ethics review board. Besides, they were paid with ¥60 per hour, which is a reasonable payment in our country.

B More Details of Entailment Detection Evaluation

When evaluating on Levy/Holt and Berant datasets, for each pair of premise and hypothesis, we search the EGs for entailment edges from the predicate of premise to the predicate of hypothesis, and return the entailment scores associated with these edges.

For entailment detection, we compare our enhanced EGs with a strong baseline BERT. On Chinese subsets, for each premise-hypothesis pair, we compute the cosine similarity between their pre-trained BERT representations of [CLS] tokens, denoted by BERT_{zh}. For the English subsets, we average the BERT hidden states of the predicate's start and end tokens as the final representations for premise or hypothesis, and calculate the cosine similarity of the representations, denoted by BERT_{en}.

C Implementation Details

The implementation details of our ATE are summarized in Table 5. Our model were trained on Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, and the training converged to be stable in 100 epochs. The training time of ATE for 100 epochs are about 43 minutes.

Hyperparameter	value
φ_1^p	0.1
φ_1^e	0.1
φ_2^p	0.3
φ_2^e	0.3
γ	1.0
k	3
$\rho^{1\to2}$	2.0
$\rho^{2\to1}$	1.2
Word Embedding Dimension	300
Learning Rate	0.001
Activation Function	ReLU
Positive v.s. Negative Ratio	1/250
Layers of GCN	2
GCN Hidden Size	300
Layers of Interaction	2
Layers of GAT in Interaction	2
GAT Hidden Size	300
Numbers of Parameters	16,736,710

Table 5: Settings for ATE.

D Impact of Available Training Data for Predicate Alignment

To explore the impact of the size of training data on our model, we compare our CGI with Glove-sim and the strongest baseline RNM by varying the proportion of seed predicate and entity alignments from 20% to 60% with a step of 10%. Figure 4 illustrates the MAP@1 for predicate alignment from English to Chinese of the three models on EGAlign dataset. As the amount of seed alignments increases, the performances of all three models gradually improve. Our CGI consistently obtains superior results compared to Glove-sim and RNM. These results show the promising performance of our model. Furthermore, according to the current trend of CGI, we believe that our model will achieve much better performance with more training data.

E Entity Alignment

Our model can also achieve accurate entity alignment simultaneously. For entity alignment, we compare with BootEA, RDGCN, HGCN-JE/JR, Glove-sim, BERT-sim as well as the state-of-the-art entity alignment model NMN (Wu et al., 2020) which presents a graph sampling method for identifying the most informative neighbors towards entity alignment and utilize a cross-graph attention-based matching mechanism to compare the neighborhood subgraphs of two entities for entity alignment.

Table 6 shows the entity alignment performance

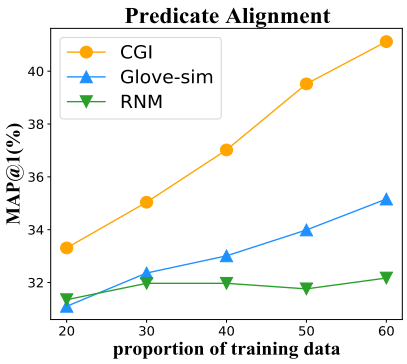

Figure 4: Predicate alignment performance of our CGI, Glove-sim and RNM when they are trained with different proportions of seed predicate and entity alignments on the EGAlign dataset.

Models	EN→ ZH		ZH → EN	
	MAP@1	MAP@10	MAP@1	MAP@10
BootEA	69.10	75.80	68.82	74.73
HGCN	78.61	82.33	76.34	79.78
RDGCN	79.67	83.40	75.52	79.36
NMN	78.16	81.49	76.32	79.87
Glove-sim	71.46	75.76	71.84	75.68
BERT-sim	63.63	67.48	63.02	66.18
CGI	**82.41**	**86.59**	**82.36**	**86.51**

Table 6: Performance on entity alignment.

on EGAlign dataset. We can observe that our CGI outperforms all the compared baselines across all the metrics, which shows that CGI can also learn better entity representations towards alignment and achieve accurate entity alignment.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation section

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Our work do not have potential risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract and section 1

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

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ 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.
No response.

C ☑ Did you run computational experiments?

section 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix B

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 5

☑ 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 6

☐ 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.)?
Not applicable. No, we do not use them.

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

Appendix A

☑ 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.?
Appendix A

☑ 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)?
Appendix A

☑ 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?
Appendix A

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Appendix A

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix A