Joint Multilingual Knowledge Graph Completion and Alignment

Vinh Tong[1], Dat Quoc Nguyen[2], Trung Thanh Huynh[3], Tam Thanh Nguyen[4], Quoc Viet Hung Nguyen[4], Mathias Niepert[1]
[1]University of Stuttgart, Germany; [2]VinAI Research, Vietnam;
[3]EPFL, Switzerland; [4]Griffith University, Australia
[1] vinh.tong@ipvs.uni-stuttgart.de, [2]v.datnq9@vinai.io

Abstract

Knowledge graph (KG) alignment and completion are usually treated as two independent tasks. While recent work has leveraged entity and relation alignments from multiple KGs, such as alignments between multilingual KGs with common entities and relations, a deeper understanding of the ways in which multilingual KG completion (MKGC) can aid the creation of multilingual KG alignments (MKGA) is still limited. Motivated by the observation that structural inconsistencies – the main challenge for MKGA models – can be mitigated through KG completion methods, we propose a novel model for jointly completing and aligning knowledge graphs. The proposed model combines two components that jointly accomplish KG completion and alignment. These two components employ relation-aware graph neural networks that we propose to encode multi-hop neighborhood structures into entity and relation representations. Moreover, we also propose (i) a structural inconsistency reduction mechanism to incorporate information from the completion into the alignment component, and (ii) an alignment seed enlargement and triple transferring mechanism to enlarge alignment seeds and transfer triples during KGs alignment. Extensive experiments on a public multilingual benchmark show that our proposed model outperforms existing competitive baselines, obtaining new state-of-the-art results on both MKGC and MKGA tasks.

1 Introduction

Knowledge graphs (KGs) represent facts about real-world entities as triples of the form (`head_entity`, `relation_type`, `tail_entity`). KGs are widely used in numerous applications and research domains such as dialogue systems (Jung et al., 2020), machine translation (Zhao et al., 2020), and electronic medical records (Rotmensch et al., 2017). Almost all KGs, however, even those at the scale of billions of triples, are far from being complete. This motivates KG completion (KGC) which aims to derive missing triples from an incomplete knowledge graph (Bordes et al., 2013; Wang et al., 2014; Yang et al., 2015; Trouillon et al., 2016; Liu et al., 2017; Dettmers et al., 2018; Nguyen, 2020; Ji et al., 2021; Tong et al., 2021).

Most popular KGs such as YAGO (Suchanek et al., 2007), BabelNet (Navigli and Ponzetto, 2010), and DBpedia (Lehmann et al., 2015), are multilingual, that is, they contain sets of triples constructed from sources in different languages. Fortunately, these KGs often complement each other since a KG in one language might be more comprehensive in some domains compared to a KG in a different language, and vice versa, while still sharing a large number of entities and relation types (Sun et al., 2020a). Especially KGs for low-resource languages could benefit from triples contained in KGs for high-resource languages. Although numerous approaches to KGC have been proposed in recent years (Bordes et al., 2013; Dettmers et al., 2018; Vashishth et al., 2020), most of these only operate on one KG at a time. Treating KGs independently, however, might lead to poor performance due to the sparseness of low-resource languages. Motivated by this observation, some methods have tried to improve multilingual KG completion (MKGC) using multilingual KG alignment (MKGA) (Chen et al., 2020; Singh et al., 2021; Huang et al., 2022).

The alignment problem is challenging due to the varying levels of completeness of monolingual KGs (Sun et al., 2020c). The resulting structural inconsistency between KGs leads to the problem of corresponding entities in two KGs having vastly different embeddings (Xu et al., 2018). Figure 1 illustrates that, in principle, MKGC and MKGA should be mutually beneficial. Given the alignment seed set $\{(E, E^*), (D, D^*), (B, B^*)\}$, the task here is to find corresponding entities for A and C which are A^* and C^*, respectively. The two KGs share a similar structure but the triple $(B^*, r4, A^*)$ in KG^*

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4646–4658
December 7-11, 2022 ©2022 Association for Computational Linguistics

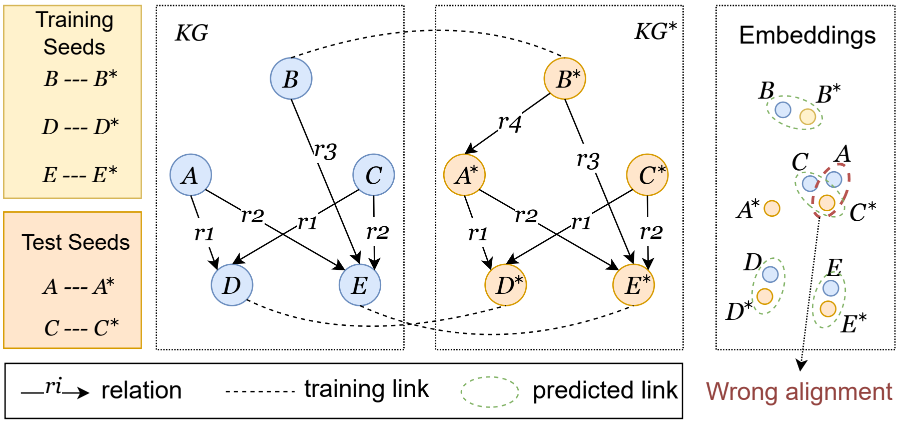

Figure 1: The incompleteness of KG (missing triple $(B, r4, A)$) might lead to a wrong alignment prediction (i.e. both A and C are predicted to be aligned to C^*). If A and A^* were aligned, however, the missing triple $(B, r4, A)$ in KG could be found by transferring triple $(B^*, r4, A^*)$ from KG^*.

has no corresponding triple in KG. This causes the embeddings of entities A and A^* to differ and, thus, makes it difficult to identify the alignment between A and A^*. Indeed, A is more likely to be aligned to C^* as they share a comparable local structure (similar degree and 1-hop neighbor set). Thus, completing missing triples is crucial to improve the alignment quality. Indeed, if one aligned A to A^*, one could recover $(B, r4, A)$ by transferring $(B^*, r4, A^*)$ from KG^*.

Motivated by these observations, we propose **JMAC**, a method for **J**oint **M**ultilingual KG **C**ompletion and **A**lignment, consisting of two interdependent Completion and Alignment components. Both components employ relation-aware graph neural networks (GNNs) to encode multi-hop neighborhood information into entity and relation embeddings. The Completion component is trained to reconstruct missing triples using the TransE translation-based loss (Bordes et al., 2013) and an additional loss term that incorporates information about the already known alignments. While we learn separate embeddings for the Alignment and Completion components, the embeddings of the Completion component are used within the Alignment component, mitigating the aforementioned problem of structural inconsistencies. In addition, we propose a mechanism for estimating the alignment entropy which is used to adaptively and iteratively grow the alignment seed set. Finally, we also propose a method for transferring triples based on the currently derived alignments.

Our contributions are as follows:

- We propose JMAC, a two-component architecture consisting of Completion and Alignment components for joint multilingual KG completion and alignment.
- We propose a relation-aware GNN for KG embeddings, which learns representations for alignment and completion tasks.
- We introduce a structural inconsistency reduction mechanism that fuses embeddings from the Completion component with those of the Alignment component.
- We propose an alignment seed enlargement and triple transferring mechanism.
- We conduct extensive experiments using the public multilingual benchmark DBP-5L (Chen et al., 2020) and show that our model outperforms existing competitive baselines and achieves state-of-the-art results on both MKGC and MKGA tasks.

2 Problem Definition and Related Work

Let $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ denote a KG, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ denote the sets of entities, relations, and triples, respectively. A triple $(e_h, r, e_t) \in \mathcal{T}$ is an atomic unit, which represents some relation $r \in \mathcal{R}$ between a head entity $e_h \in \mathcal{E}$ and a tail entity $e_t \in \mathcal{E}$.

2.1 Multilingual KG completion (MKGC)

Given a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, the KG completion (KGC) task aims to predict missing triples (e_h, r, e_t), that is, to predict the missing tail entity $e_t \in \mathcal{E}$ of an incomplete triple $(e_h, r, ?)$ or the missing head entity $e_h \in \mathcal{E}$ of an incomplete triple $(?, r, e_t)$, where ? denotes the missing element.

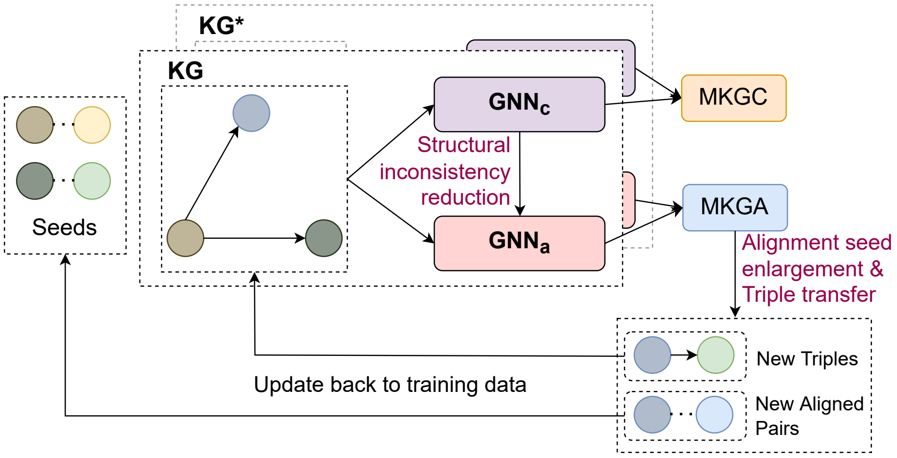
KG*
KG
Seeds
GNN$_c$
GNN$_a$
Structural inconsistency reduction
MKGC
MKGA
Alignment seed enlargement & Triple transfer
New Triples
New Aligned Pairs
Update back to training data

Figure 2: The overall architecture of JMAC.

Embedding models for KG completion have been proven to give state-of-the-art results, representing entities and relation types with latent feature vectors, matrices, and/or third-order tensors (Nguyen, 2020; Ji et al., 2021). These models define a score function f and are trained to make the score $f(e_h, r, e_t)$ of a correct triple (e_h, r, e_t) larger than the score $f(e_{h'}, r', e_{t'})$ of an incorrect or not known to be correct triple $(e_{h'}, r', e_{t'})$. The earliest instances of these embedding models use shallow neural networks with translation-based score functions (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015). Recently, KGC approaches using deep embedding models and more complex scoring functions have been proposed, such as CNN-based models (Dettmers et al., 2018; Nguyen et al., 2018), RNN-based models (Liu et al., 2017; Guo et al., 2018), and GNN-based models (Schlichtkrull et al., 2018; Shang et al., 2019; Vashishth et al., 2020; Nguyen et al., 2022).

The MKGC task is to perform the KGC task on a KG given the availability of other KGs in different languages (Chen et al., 2020; Huang et al., 2022).

2.2 Multilingual KG alignment (MKGA)

MKGA, which is also known as cross-lingual entity alignment, aims to match entities with their counterparts from KGs in different languages (Chen et al., 2017; Wang et al., 2018; Wu et al., 2019; Sun et al., 2020b). Without loss of generalization, we define the alignment between a source graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ and a target graph $\mathcal{G}^* = (\mathcal{E}^*, \mathcal{R}^*, \mathcal{T}^*)$. For each entity $e \in \mathcal{E}$, the MKGA task is now to find $e^* \in \mathcal{E}^*$ (if any).

Existing models compute an alignment matrix whose elements represent the similarity score between any two entities $e \in \mathcal{E}$ and $e^* \in \mathcal{E}^*$ across two KGs. The models then employ a greedy matching algorithm (Kollias et al., 2011) to infer matching entities from the alignment matrix. The models typically require an alignment seed set $\mathbb{L}$ of pre-aligned entity pairs (e, e^*).

2.3 Joint MKGC and MKGA

The joint MKGC and MKGA problem aims to infer both, new triples for each KG and new aligned entity pairs for each pair of KGs. Performing two tasks jointly might be beneficial: missing triples (e_h, r, e_t) in one KG could be recovered by cross-checking another KG via the alignment, which, in turn, could be boosted by the newly added triples.

Despite the obvious benefit to complete and align KGs jointly, there has not been much work addressing the problem. A notable exception is the application of multi-task learning to the problem (Singh et al., 2021). The proposed multi-task model, however, is not able to capture local neighborhood information. Another limitation is the missing robustness to the previously described issue of structural inconsistencies between the KGs during training.

3 JMAC: Joint Multilingual Alignment and Completion

Figure 2 illustrates the architecture of **JMAC** which consists of the Completion and Alignment components, respectively. Each component uses a relation-aware graph neural network (GNN) to encode multi-hop neighborhood information into entity and relation embeddings. The use of two GNN encoders is beneficial as embeddings that are suitable for the alignment might differ from those that are most beneficial for the completion problem.

3.1 Relation-aware graph neural network

To better capture relation information, we propose a GNN architecture that uses relation-aware messages and relation-aware attention scores.

We unify heterogeneous information from KGs using a GNN with K layers. For the k-th layer (denoted by the superscript ^{k}), we update the representation $\mathbf{a}_e^{k+1}$ of each entity $e \in \mathcal{E}$ as:

$$\mathbf{a}_e^{k+1} = g\left(\sum_{(e',r)\in\mathcal{N}(e)} \alpha_{e,e',r}^k \mathbf{m}_{e',r}^k + \mathbf{a}_e^k\right) \tag{1}$$

where $\mathbf{a}_e^k \in \mathbb{R}^n$ is the vector representation of entity e at the k-th layer; $\mathcal{N}(e) = \{(e',r)|(e,r,e') \in \mathcal{T} \cup (e',r,e) \in \mathcal{T}\}$ is the neighbor set of entity e; and the vector $\mathbf{m}_{e',r}^k \in \mathbb{R}^n$ denotes the message passed from neighbor entity e' to entity e through relation r. Here, $\alpha_{e,e',r}^k$ represents the attention weight that regulates the importance of the message $\mathbf{m}_{e',r}^k$ for entity e; and $g(.)$ is a linear transformation followed by a Tanh function.

The innovations of our GNN-based model, which we describe in more detail in the following two sections, are (i) we make the message-passing NN to be relation-aware by learning the relation embedding $\mathbf{a}_r^k \in \mathbb{R}^n$ for each relation $r \in \mathcal{R}$ and by integrating it into the entity message passing scheme $\mathbf{m}_{e',r}^k$ and (ii) we introduce an attention weight $\alpha_{e,e',r}^k$ to further enhance the relation-aware capability of our GNN-based embeddings.

Relation-aware message Unlike existing GNN-based approaches that infer relation embeddings from learned entity embeddings (Sun et al., 2020b), our approach allows entity and relation embeddings to be learned jointly and, thus, to both contribute to the message passing neural network. This is achieved through an entity-relation composition operation. The message $\mathbf{m}_{e',r}^k$ in Equation 1 is defined as:

$$\mathbf{m}_{e',r}^k = \mathbf{a}_{e'}^k - \mathsf{MLP}_{\text{comp}}^k\left(\mathbf{a}_r^k\right) \tag{2}$$

where $\mathsf{MLP}_{\text{comp}}^k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a two-layer MLP with the $\mathsf{LeakyReLU}$ activation function.

Here, the relation embedding is updated by

$$\mathbf{a}_r^{k+1} = \mathsf{MLP}_{\text{rel}}^k\left(\mathbf{a}_r^k\right) \tag{3}$$

where $\mathsf{MLP}_{\text{rel}}^k : \mathbb{R}^n \rightarrow \mathbb{R}^n$ maps relations to a new embedding space and allows them to be utilized in the next layer.

Relation-aware attention We define the weight $\alpha_{e,e',r}^k$ in Equation 1 to be relation-aware as:

$$\alpha_{e,e',r}^k = \frac{\exp\left(\mathsf{MLP}_{\text{att}}^k\left(\mathbf{a}_e^k \circ \mathbf{m}_{e',r}^k\right)\right)}{\sum\limits_{(e",r")\in\mathcal{N}(e)} \exp\left(\mathsf{MLP}_{\text{att}}^k\left(\mathbf{a}_e^k \circ \mathbf{m}_{e",r"}^k\right)\right)} \tag{4}$$

where $\mathsf{MLP}_{\text{att}}^k : \mathbb{R}^{2\times n} \rightarrow \mathbb{R}$; and $\circ$ denotes the vector concatenation operator. As the message vector $\mathbf{m}_{e',r}^k$ contains the information of not only neighbor entity e' but also neighbor relation r, our attentive score $\alpha_{e,e',r}^k$ can capture the importance of the message coming from entity e' to entity e conditioned on relation r connecting them.

Notation extension Recall that we use two different relation-aware GNN encoders as illustrated in Figure 2: one for the Completion and one for the Alignment component. To distinguish these two encoders, we use $\mathbf{c}$ and $\mathbf{a}$ to denote the embedding representations used for the Completion and Alignment components, respectively. In particular, $\mathbf{a}_e^k$ and $\mathbf{a}_r^k$ are now the corresponding embeddings of entity e and relation r at the k-th layer of the relation-aware GNN in the Alignment component. Furthermore, $\mathbf{c}_e^k$ and $\mathbf{c}_r^k$ are the corresponding embeddings of entity e and relation r at the k-th layer of the relation-aware GNN in the Completion component (computed as those of the Alignment component defined in equations 1 and 3).

3.2 Completion component

The Completion component works similarly to KG embedding models (Nguyen, 2020; Ji et al., 2021) which compute a score $f(e_h, r, e_t)$ for each triple (e_h, r, e_t). Our score function f is based on TransE (Bordes et al., 2013) and is computed across all hidden layers of the relation-aware GNN encoder in the Completion component as follows:

$$f^k(e_h, r, e_t) = -\|\mathbf{c}_{e_h}^k + \mathbf{c}_r^k - \mathbf{c}_{e_t}^k\|_{\ell_1} \tag{5}$$

$$f(e_h, r, e_t) = \sum_k f^k(e_h, r, e_t) \tag{6}$$

We use a margin-based pairwise ranking loss (Bordes et al., 2013) across all hidden layers:

$$\mathcal{L}_{c\_1} = \sum_{\substack{k \\ (e_h,r,e_t)\in\mathcal{T} \\ (\overline{e}_h,r,\overline{e}_t)\in\overline{\mathcal{T}}}} [\gamma_c - f^k(e_h, r, e_t) + f^k(\overline{e}_h, r, \overline{e}_t)]_+ \tag{7}$$

where $[x]_+ = \max(0, x)$; $\gamma_c > 0$ is the margin hyper-parameter; and $\overline{\mathcal{T}}$ is the set of incorrect triples constructed by corrupting either the head or the tail entity the correct triple $(e_h, r, e_t) \in \mathcal{T}$.

Also, given the availability of the alignment seed set $\mathbb{L}$ of pre-aligned entity pairs (e, e^*) among KGs as mentioned in Section 2.2, we additionally compute the following alignment constraint loss:

$$\mathcal{L}_{c\_2} = \sum_{k} \sum_{(e,e^*)\in\mathbb{L}} \mathsf{d_{cos}}\left(\mathbf{c}_e^k, \mathbf{c}_{e^*}^k\right) \tag{8}$$

where $\mathsf{d_{cos}}$ denotes the cosine distance.

To incorporate alignment information into the Completion component, our MKGC loss is computed as the sum of the two losses $\mathcal{L}_{c\_1}$ and $\mathcal{L}_{c\_2}$:

$$\mathcal{L}_c = \mathcal{L}_{c\_1} + \mathcal{L}_{c\_2} \tag{9}$$

3.3 Alignment component

The Alignment component is to perform the MKGA task as defined in Section 2.2. We propose to incorporate a structural inconsistency reduction inherited from the Completion into the Alignment component. We also introduce a mechanism to enlarge the alignment seed set $\mathbb{L}$.

Structural inconsistency reduction (SIR) The different completeness levels of KGs lead to structural inconsistencies that might cause incorrect alignment predictions. Fortunately, entity and relation embeddings in the Completion component (i.e. $\mathbf{c}_e^k$ and $\mathbf{c}_r^k$) could help reconstruct missing triples, reducing the structural inconsistencies between KGs. To this end, we propose to incorporate the Completion component embeddings at the k-th layer of the relation-aware GNN in the Alignment component as follows:

$$\mathbf{a}_e^k = \mathsf{MLP}_{\text{a\_1}}^k\left(\mathbf{c}_e^k \circ \mathbf{a}_e^k\right) \quad \forall e \tag{10}$$

$$\mathbf{a}_r^k = \mathsf{MLP}_{\text{a\_2}}^k\left(\mathbf{c}_r^k \circ \mathbf{a}_r^k\right) \quad \forall r \tag{11}$$

where $\mathsf{MLP}_{\text{a\_1}}^k : \mathbb{R}^{2\times n} \rightarrow \mathbb{R}^n$; and $\mathsf{MLP}_{\text{a\_2}}^k : \mathbb{R}^{2\times n} \rightarrow \mathbb{R}^n$. The transformed embeddings then will be used as input for the next layer following equations 1 and 3. This allows the structural inconsistency reduction to take place at every GNN layer of the two components, enabling their deep integration.

Final entity and relation embeddings for MKGA We compute the final embeddings for entities and relations for this Alignment component as follows:

$$\mathbf{a}_e = \mathsf{MLP}_{\text{a\_3}}\left(\mathbf{a}_e^0 \circ \mathbf{a}_e^1 \circ ... \circ \mathbf{a}_e^K\right) \tag{12}$$

$$\mathbf{a}_r = \mathsf{MLP}_{\text{a\_4}}\left(\mathbf{a}_r^0 \circ \mathbf{a}_r^1 \circ ... \circ \mathbf{a}_r^K\right) \tag{13}$$

where $\mathsf{MLP}_{\text{a\_3}} : \mathbb{R}^{(K+1)\times n} \rightarrow \mathbb{R}^n$; and $\mathsf{MLP}_{\text{a\_4}} : \mathbb{R}^{(K+1)\times n} \rightarrow \mathbb{R}^n$.

Alignment seed enlargement and Triple transferring (EnTr) As mentioned in Section 2.2, existing alignment models require an alignment seed set $\mathbb{L}$ of pre-aligned entity pairs for training. An intuitive approach to improve alignment results is to iteratively increase the size of $\mathbb{L}$ during training. The number of alignments by which the size of $\mathbb{L}$ is increased should depend on some notion of certainty the Alignment component has in its predictions. For example, in the early stages of the alignment process, when the KGs are still sparsely connected, $\mathbb{L}$ should be smaller. The more confident the Alignment component is in its predictions, the larger should $\mathbb{L}$ be. Hence, we propose EnTr, a method to enlarge the alignment seed set. EnTr estimates an optimal number of new entity pairs to be added to $\mathbb{L}$ according to a measure of alignment certainty. At each training epoch, EnTr first computes an alignment matrix $\mathbf{A}$, where each element is the cosine similarity between any two entity embeddings of the two KGs, that is, $\mathbf{A}(e, e^*) = 1 - \mathsf{d_{cos}}(\mathbf{a}_e, \mathbf{a}_{e^*})$. EnTr then computes the Shannon entropy of the softmax distribution over this alignment matrix:

$$\mathsf{P}(e^*|e) = \frac{\exp\left(\mathbf{A}(e, e^*)\right)}{\sum_{e^\star\in\mathcal{E}^*} \exp\left(\mathbf{A}(e, e^\star)\right)} \tag{14}$$

$$\mathsf{H}(\mathbf{A}) = -\sum_{e\in\mathcal{E}} \sum_{e^*\in\mathcal{E}^*} \mathsf{P}(e^*|e) \log \mathsf{P}(e^*|e) \tag{15}$$

Since the entropy measures the uncertainty of the alignment predictions – lower entropy corresponds to a smaller alignment uncertainty – EnTr implements an uncertainty-mediated estimation of the size q of $\mathbb{L}$ as follows:

$$\mathsf{q} = \left\lfloor \beta \cdot \frac{\mathsf{H}(\widetilde{\mathbf{A}}) - \mathsf{H}(\mathbf{A})}{\mathsf{H}(\widetilde{\mathbf{A}})} \cdot \min(|\mathcal{E}|, |\mathcal{E}^*|) \right\rfloor \tag{16}$$

where $\widetilde{\mathbf{A}}$ is the alignment matrix before training and $\beta \in [0, 1]$ is a hyper-parameter controlling the number of new entity pairs to generate. EnTr then chooses the q entity pairs with the q highest cosine similarity scores from $\mathbf{A}$ and sets $\mathbb{L}$ to contain these entity pairs.

EnTr also performs transfer of triples between

Language	Greek	Japanese	French	Spanish	English
#Relation	111	128	144	178	831
#Entity	5,231	11,805	12,382	13,176	13,996
#Triple	13,839	28,744	54,066	49,015	80,167

Table 1: Statistics of DBP-5L.

the KGs that logically follow from the existing alignments. In particular, for each two aligned entity pairs (e, e^*) and (e', e'^*), if r is a relation connecting e and e', EnTr connects e^* and e'^* by r as well. This allows the two KGs' structures to become gradually more similar over time.

Optimization The learning objective is to minimize the distance between correctly aligned entity pairs while maximizing the distance between negative entity pairs using a margin-based pairwise ranking loss:

$$\mathcal{L}_a = \sum_{\substack{(e,e^*)\in\mathbb{L} \\ (\bar{e},\bar{e}^*)\in\bar{\mathbb{L}}}} [\gamma_a + \mathsf{d}_{\mathsf{cos}}(\mathbf{a}_e, \mathbf{a}_{e^*}) - \mathsf{d}_{\mathsf{cos}}(\mathbf{a}_{\bar{e}}, \mathbf{a}_{\bar{e}^*})]_+ \tag{17}$$

where $\gamma_a > 0$ is the margin hyper-parameter; and $\bar{\mathbb{L}}$ is the set of negative entity pairs, which is constructed by replacing one entity of each correctly aligned pair by its nearest entities (Wu et al., 2019).

3.4 Model training

We optimize the losses $\mathcal{L}_c$ and $\mathcal{L}_a$ iteratively, using two optimizers respectively for the Completion and Alignment losses. In particular, we hold the Alignment component's parameters fixed and optimize only the loss $\mathcal{L}_c$. Then we hold the Completion component's parameters fixed and only optimize the loss $\mathcal{L}_a$. We keep iterating this process in each training epoch.

4 Experimental Setup

4.1 Dataset

Following previous work (Singh et al., 2021; Huang et al., 2022), we conduct experiments using the benchmark DBP-5L[1] (Chen et al., 2020), publicly available for both MKGC and MKGA tasks.[2] DBP-5L consists of 1,392 relations, 56,590 entities, and 225,831 triples across the five languages Greek (**EL**), Japanese (**JA**), French (**FR**), Spanish (**ES**), and English (**EN**). Table 1 presents statistics for each DBP-5L language. Here, each language is referred to as a KG. The DBP-5L benchmark is created for MKGC evaluation. However, each language pair also has pre-aligned entity pairs as the alignment seeds. In particular, on average, about 40% of the entities in each KG have their counterparts at other KGs. Thus, it can also be used for MKGA evaluation (Singh et al., 2021). Similar to prior works, we use the same split of training, validation, and test data, for MKGC available in DBP-5L for each KG. For MKGA, we use the same 50-50 split of the alignment seeds for training and test, as used in AlignKGC (Singh et al., 2021).[3]

4.2 Evaluation protocol

For MKGC, and following previous work (Chen et al., 2020; Singh et al., 2021; Huang et al., 2022), each correct test triple (e_h, r, e_t) is corrupted by replacing the tail entity e_t with each of the other entities in turn, and then the correct test triple and corrupted ones are ranked in descending order of their score. Similar to the previous work, before ranking, we also applied the "Filtered" setting protocol (Bordes et al., 2013). We employ standard evaluation metrics, including the mean reciprocal rank (**MRR**), **Hits@1** (i.e. the proportion of correct test triples that are ranked first) and **Hits@10** (i.e. the proportion of correct test triples that are ranked in the top 10 predictions). Here, a higher score reflects better prediction result.

For MKGA, and following previous work (Chen et al., 2017; Wu et al., 2019; Sun et al., 2020b; Singh et al., 2021), each correct test pair (e, e^*), where $e \in \mathcal{E}$ and $e^* \in \mathcal{E}^*$, is corrupted by replacing entity e^* with each of the other entities from $\mathcal{E}^*$ in turn, and then the correct test pair and corrupted ones are ranked in descending order of their similarity score. We also employ the evaluation metrics MRR, Hits@1 and Hits@10.

[1] `https://github.com/stasl0217/KEnS/tree/main/data`

[2] Huang et al. (2022) create another multilingual benchmark named E-PKG (to be released at `https://github.com/amzn/ss-aga-kgc`), however, it is not yet available as of 24/06/2022, EMNLP 2022's submission deadline.

[3] JMAC can perform KGA on a benchmark that is purely constructed for the KGA task. We show state-of-the-art results obtained for JMAC on a KGA benchmark in the Appendix.

Method	Align.	Greek			Japanese			French			Spanish			English		
		H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H10	MRR	H1	H@10	MRR
TransE	0%	13.1	43.7	24.3	21.1	48.5	25.3	13.5	45.0	24.4	17.5	48.8	27.6	7.3	29.3	16.9
RotatE	0%	14.5	36.2	26.2	26.4	60.2	39.8	21.2	53.9	33.8	23.2	55.5	35.1	12.3	30.4	20.7
KG-BERT	0%	17.3	40.1	27.3	26.9	59.8	38.7	21.9	54.1	34.0	23.5	55.9	35.4	12.9	31.9	21.0
KenS	100%	26.4	66.1	-	32.9	64.8	-	22.3	60.6	-	25.2	62.6	-	14.4	39.6	-
SS-AGA	100%	30.8	58.6	35.3	34.6	66.9	42.9	25.5	61.9	36.6	27.1	65.5	38.4	16.3	41.3	23.1
AlignKGC w/o SI	50%	55.1	84.2	65.5	46.7	74.4	56.6	44.5	74.0	54.9	44.0	71.4	53.4	28.5	54.9	37.5
AlignKGC w/ SI	50%	**58.2**	88.6	69.4	49.3	78.7	60.1	48.4	79.4	59.5	**48.0**	76.6	58.0	**31.7**	59.8	41.3
JMAC w/o SI	50%	46.8	90.3	62.3	48.1	85.2	61.3	43.8	83.8	58.0	35.6	76.7	50.3	24.5	65.3	38.3
JMAC w/ SI	50%	55.2	**97.5**	**71.7**	**53.3**	**91.4**	**66.8**	**49.3**	**91.3**	**64.5**	45.4	**88.2**	**61.0**	29.5	**72.7**	**44.6**

Table 2: MKGC results. All metrics are reported in %. Here, H@1 and H@10 abbreviate Hits@1 and Hits@10, respectively. "Align." denotes the percentage of alignment seeds used by each model when training. The first three models are monolingual baselines (i.e. equivalent to the "Align." rate of 0%), while the remaining models are multilingual ones. KenS (Chen et al., 2020) and SS-AGA (Huang et al., 2022) are proposed for MKGC only, employing all alignment seeds, i.e. their "Align." rate is 100%. Results of TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), KG-BERT (Yao et al., 2019), KenS and SS-AGA are taken from Huang et al. (2022), that are reported only with surface information (w/ SI). Results of AlignKGC (Singh et al., 2021) are taken from Singh et al. (2021).

4.3 Implementation details

The availabilty of surface information (**SI**) such as entity names makes the alignment problem less challenging (Xiang et al., 2021). When surface information is not used by the methods (denoted as **w/o SI**), the entity and relation embeddings are randomly initialized. When surface information is used (denoted as **w/ SI**), initial entity and relation embeddings are obtained from pre-trained text embedding models (Singh et al., 2021; Huang et al., 2022). Therefore, we also evaluate these two problems and refer to them as **JMAC w/o SI** and **JMAC w/ SI**.

We implement our model using Pytorch (Paszke et al., 2019). We iteratively train our JMAC components up to 30 epochs with two Adam optimizers (Kingma and Ba, 2014). We use a grid search to choose the number of GNN hidden layers $K \in \{1, 2, 3\}$, the initial Adam learning rates $\lambda \in \left\{1e^{-4}, 5e^{-4}, 1e^{-3}\right\}$, the controllable hyper-parameter $\beta \in \{0.1, 0.2, 0.3\}$ from Equation 16, the margin hyper-parameters γ_c and γ_a $\in \{0, 5, 10\}$, and the input dimension and MLP hidden sizes $n \in \{128, 256, 512\}$. The test set results for the two tasks are reported for the model checkpoint which obtains the highest MRR on the validation set of the MKGC task.

5 Main Results

5.1 MKGC results

Comparison results Table 2 lists MKGC results for JMAC and other strong baselines on the DBP-5L test sets. Overall, the multilingual models perform better than the monolingual ones. It is not surprising that JMAC and AlignKGC without surface information (w/o SI) obtain lower numbers than their counterparts using SI (w/ SI). For example, with SI information, JMAC achieves an average improvement of about 8% points for Hits@10. Note that, our "JMAC w/o SI" still produces higher Hits@10 results than "AlignKGC w/ SI" on all KGs, and in general, performs better than "AlignKGC w/o SI". Compared to SS-AGA that uses both SI and all available alignment seeds for training, our "JMAC w/o SI" Hits@10 results are about 30% better for Greek, 20% better for Japanese, French and English, and 10% better for Spanish. We find that our "JMAC w/ SI" obtains the highest results across all KGs on all evaluation metrics, producing new state-of-the-art performances (except the second-highest Hits@1 scores on Greek, Spanish and English where "AlignKGC w/ SI" produces the highest Hits@1 scores).

Ablation study Table 3 presents ablation results. Removing or replacing any component or mechanism decreases the model's performance. The largest decrease is observed without the use of the Alignment component (w/o Align.) where the MRR scores drop about 15% points on average. The model also performs substantially poorer when it is not using the alignment seed enlargement and triple transferring mechanism (w/o EnTr). Here, the model's Hits@10 scores decrease about 10% in Greek, Japanese, French and Spanish. The performance drops incurred by using the same relation-aware GNN encoder (w/ 1-GNN) or not using

Variants	Greek			Japanese			French			Spanish			English		
	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
JMAC w/ SI	**55.2**	**97.5**	**71.7**	**53.3**	**91.4**	**66.8**	**49.3**	**91.3**	**64.5**	**45.4**	**88.2**	**61.0**	**29.5**	**72.7**	**44.6**
(i) w/o RA-GNN	53.3	95.3	70.6	49.1	90.7	63.8	46.4	88.7	61.0	43.6	85.4	59.3	27.1	71.1	42.3
(ii) w/ 1-GNN	51.1	92.4	64.2	49.1	90.3	63.1	46.1	87.1	60.8	41.4	82.4	56.1	27.2	68.3	41.6
(iii) w/o SIR	54.8	96.8	71.1	50.9	91.1	64.3	48.2	90.3	63.6	44.9	86.3	60.5	28.4	71.4	43.3
(iv) w/o EnTr	41.5	87.1	57.3	39.7	80.6	54.4	40.5	80.3	54.2	36.3	76.3	50.5	27.2	67.3	39.3
(v) w/o Align.	35.3	83.4	52.1	35.3	75.8	48.8	35.3	76.8	49.9	29.3	69.1	46.3	23.5	55.8	34.8

Table 3: Ablation study for the MKGC task. **(i) w/o RA-GNN**: Without using the relation-aware message (Equation 2) and relation-aware attention (Equation 4), here we replace our proposed RA-GNNs by the graph isomorphism networks (Xu et al., 2019). **(ii) w/ 1-GNN**: Both the Completion and Alignment components share the same relation-aware GNN encoder, also leading to "w/o SIR". **(iii) w/o SIR**: Without using the structural inconsistency reduction mechanism described in Section 3.3, i.e. equations 10 and 11 are not used. **(iv) w/o EnTr**: Without using the alignment seed enlargement and triple transferring mechanism described in Section 3.3. **(v) w/o Align.**: Model variant containing only the Completion component without the Alignment one.

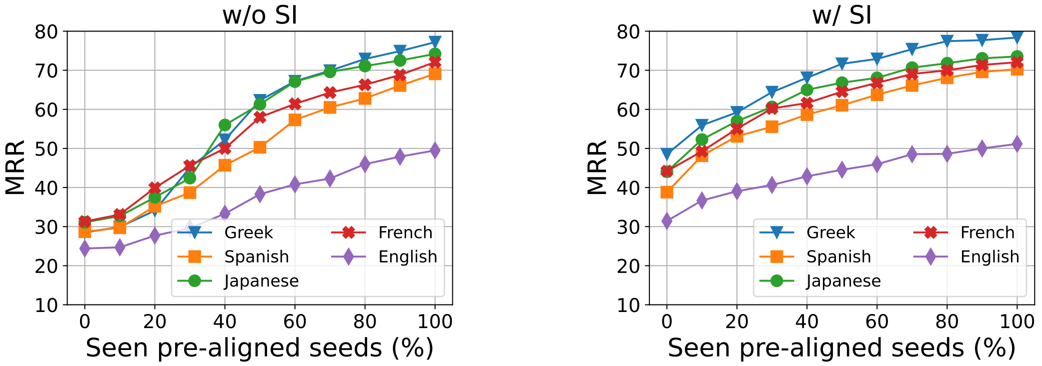

Figure 3: MKGC MRR results w.r.t different sampling percentages of alignment seed pairs.

relation-aware messages and attention (w/o RA-GNN) also demonstrate the importance of the two-component architecture design and the relation-ware message passing scheme. Although the structural inconsistency reduction mechanism aims to improve the alignment performance, we find that without it (w/o SIR), the MKGC performance also declines. This shows that the improvement in the MKGA task can lead to direct improvements in the MKGC task.

Impact of alignment seeds To have a better insight into how much MKGA can aid MKGC, we evaluate the MKGC task when using alignment seeds with a sampling percentage ranging from 0% to 100% (i.e. using all pre-aligned entity pairs). Figure 3 shows the MRR scores obtained for this experiment. Overall, our JMAC is improved when using more alignment seeds. The surface information (SI), e.g. entity names, provides informative clues about the similarity between entities across KGs (e.g. two entities with similar names are more likely to be an alignment pair). It is therefore not surprising that "JMAC w/ SI" performs better than "JMAC w/o SI" in all scenarios, especially when the used sampling percentage of alignment seeds is small (i.e. SI becomes more valuable). In addition, the completion performance gradually closer as the sampling percentage approaches 100%. The reason is possibly that when the set of used alignment seeds is large enough, there is not much for surface information to contribute to the alignment performance, which aids the completion performance.

5.2 MKGA results

Comparison results Table 4 presents the overall results of different models on the MKGA task. We refer the reader to the Appendix for results on each language pair. Overall, JMAC performs the best in both the "w/ SI" and "w/o SI" categories. Although SS-AGA performs well in the MKGC task, it performs much worse on the alignment task. Specifically, although using SI, it produces the lowest Hits@10. AlignKGC, on the other hand, achieves the third-best results in both categories. However, it is still outperformed by pure KGA models such as AliNet (obtaining 11.1% points higher Hits@1 than AlignKGC in the "w/o SI" cat-

Method	Overall		
	Hits@1	Hits@10	MRR
w/o SI			
AlignKGC	50.2	65.4	-
MTransE	28.2	44.0	36.2
AliNet	61.3	73.2	60.8
JMAC	**63.8**	**75.8**	**70.3**
w/ SI			
AlignKGC	84.8	91.9	-
SS-AGA	34.1	40.1	37.4
PSR	77.2	88.4	81.2
RDGCN	89.3	94.9	91.9
JMAC	**93.4**	**97.5**	**95.1**

Table 4: MKGA results. Results for AlignKGC are taken from Singh et al. (2021). We report our results for other baselines including MTransE (Chen et al., 2017), AliNet (Sun et al., 2020b), SS-AGA, PSR (Mao et al., 2021) and RDGCN (Wu et al., 2019), employing their publicly released implementations. See the Appendix for the training protocols of these baselines. SS-AGA is originally proposed and evaluated for MKGC only. However, it also computes and defines an alignment matrix, thus we can also evaluate SS-AGA for the MKGA task using this matrix.

Variants	Overall		
	Hits@1	Hits@10	MRR
JMAC w/ SI	**93.4**	**97.5**	**95.1**
(i) w/o RA-GNN	89.3	92.2	90.9
(ii) w/ 1-GNN	62.4	65.4	64.0
(iii) w/o SIR	91.3	94.3	92.6
(iv) w/o EnTr	92.9	95.6	94.5
(v) w/o Comple.	91.3	94.3	92.6

Table 5: Ablation study for the MKGA task. **(v) w/o Comple.**: Model variant containing only the Alignment component without the Completion one.

egory) and RDGCN (obtaining 4.5% points higher Hits@1 than AlignKGC in the "w/ SI" category).

Ablation study Table 5 shows the ablation results on the MKGA task. The model performance drops by about 5.3% points in Hits@10 without the relation-aware messages and attention (w/o RA-GNN), confirming that the relation-aware mechanism is a crucial part of our model. Using only one GNN encoder (w/ 1-GNN) for both Completion and Alignment components performs worse. This indicates that combining the objective functions and using the same feature for multiple tasks might not be optimal. The EnTr mechanism improves the model Hits@10 by about 2% points (w/o EnTr 95.6% → 97.5%). In the absence of SIR (w/o SIR), the alignment performance suffers a drop in each evaluation metric (2.1% points for Hits@1, 3.2% points for Hits@10, and 2.5% points for MRR). This indicates that the structural inconsistency reduction improves alignment performance. As the Completion component only aids the Alignment component through the SIR mechanism, the model variant without Completion component (w/o Comple.) has the same alignment performance as the model variant "w/o SIR".

6 Conclusion

We proposed JMAC, a method for joint multilingual knowledge graph completion and alignment. JMAC consists of a Completion and Alignment component and uses a new class of relation-aware GNNs for learning entity and relation embeddings suitable for both the completion and alignment tasks. We also propose a structural inconsistency reduction mechanism that fuses entity and relation embeddings from the Completion component with those of the Alignment component. We also introduce another mechanism to enlarge alignment seeds and transfer triples among KGs during training. Extensive experiments using the benchmark DBP-5L (Chen et al., 2020) show that JMAC performs better than previous strong baselines, producing state-of-the-art results. We publicly release the implementation of our JMAC at `https://github.com/vinhsuhi/JMAC`.

Limitations

Our experimental results demonstrate that our JMAC model effectively solves the structural inconsistency problem. However, our model works based on the assumption that the surface information (SI), i.e. entity name, of an entity across different KG languages is the same or similar. In fact, some entities might have very different SI compared to their versions across different KGs, due to incorrect annotations during KGs construction or different description caused by the language barrier. We refer this issue to as SI inconsistency. When the level of SI inconsistency is high, using SI information might have reversed impacts on the model performance. A robust model should be able to decide how much it can rely on the SI. Our model has no mechanism to perform that at the moment.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *NIPS*.

Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2017. Multilingual Knowledge Graph Embeddings for Cross-Lingual Knowledge Alignment. In *IJCAI*.

Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, and Carlo Zaniolo. 2020. Multilingual Knowledge Graph Completion via Ensemble Knowledge Transfer. In *Findings of EMNLP*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI*.

Lingbing Guo, Qingheng Zhang, Weiyi Ge, Wei Hu, and Yuzhong Qu. 2018. DSKG: A deep sequential model for knowledge graph completion. In *CCKS*.

Zijie Huang, Zheng Li, Haoming Jiang, Tianyu Cao, Hanqing Lu, Bing Yin, Karthik Subbian, Yizhou Sun, and Wei Wang. 2022. Multilingual Knowledge Graph Completion with Self-Supervised Adaptive Graph Alignment. In *ACL*.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2021. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. *TNNLS*.

Jaehun Jung, Bokyung Son, and Sungwon Lyu. 2020. AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue. In *EMNLP*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint, arXiv:1412.6980*.

Giorgos Kollias, Shahin Mohammadi, and Ananth Grama. 2011. Network Similarity Decomposition (NSD): A Fast and Scalable Approach to Network Alignment. *TKDE*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *SWJ*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*.

Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-Relational Embeddings. In *ICML*.

Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Are Negative Samples Necessary in Entity Alignment? An Approach with High Performance, Scalability and Robustness. In *CIKM*.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a Very Large Multilingual Semantic Network. In *ACL*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *NAACL*.

Dai Quoc Nguyen, Vinh Tong, Dinh Phung, and Dat Quoc Nguyen. 2022. Node Co-occurrence based Graph Neural Networks for Knowledge Graph Link Prediction. In *WSDM*.

Dat Quoc Nguyen. 2020. A survey of embedding models of entities and relationships for knowledge graph completion. In *TextGraphs*.

Adam Paszke, Sam Gross, Francisco Massa, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. 2017. Learning a Health Knowledge Graph from Electronic Medical Records. *Scientific Reports*.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC*.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion. In *AAAI*.

Harkanwar Singh, Prachi Jain, Mausam, and Soumen Chakrabarti. 2021. Multilingual Knowledge Graph Completion with Joint Relation and Entity Alignment. In *AKBC*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A Core of Semantic Knowledge. In *WWW*.

Jian Sun, Yu Zhou, and Chengqing Zong. 2020a. Dual Attention Network for Cross-lingual Entity Alignment. In *COLING*.

Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020b. Knowledge Graph Alignment Network with Gated Multi-hop Neighborhood Aggregation. In *AAAI*.

Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020c. A Benchmarking Study of Embedding-Based Entity Alignment for Knowledge Graphs. In *VLDB*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *ICLR*.

Vinh Tong, Dai Quoc Nguyen, Dinh Phung, and Dat Quoc Nguyen. 2021. Two-view Graph Neural Networks for Knowledge Graph Completion. *arXiv preprint*, arXiv:2112.09231.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *ICLR*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*.

Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks. In *EMNLP*.

Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In *IJCAI*.

Yuejia Xiang, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Zhenxi Lin, and Yefeng Zheng. 2021. OntoEA: Ontology-guided Entity Alignment via Joint Knowledge Graph Embedding. In *Findings of ACL*.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks? In *ICLR*.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *ICML*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *ICLR*.

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

Yang Zhao, Lu Xiang, Junnan Zhu, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2020. Knowledge Graph Enhanced Neural Machine Translation via Multi-task Learning on Sub-entity Granularity. In *COLING*.

A Appendix

A.1 Training protocols for baselines

For the baseline alignment models listed in Table 4, we apply the same training protocol as detailed in Section 4.3 w.r.t. the optimizer, the hidden layers, the initial learning rate values and the number of training epochs. For GNN-based alignment models AliNet, SS-AGA, PSR and RDGCN, we also search their number of GNN layers from $\{1, 2, 3\}$. For their other hyper-parameters, we use their implementation's default values.

A.2 MKGA results on the DBP-5L dataset

Tables 6–11 detail alignment results of experimental models as well as ablation results of our JMAC for all language pairs in the DBP-5L dataset.

A.3 Additional results for the KGA task

Note that our JMAC can perform KGA on a benchmark that is purely constructed for the KGA task. To further demonstrate the effectiveness of our JMAC, we conduct an additional KGA experiment using bilingual KG pairs from the OpenEA 15K benchmark DBP1.0 (Sun et al., 2020c). Each KG pair consists of two versions V1 and V2 which are the sparse and dense ones, respectively. The alignment seeds are divided into 20%, 10%, and 70% for training, validation and test, respectively. Statistics of the bilingual KG pairs from the OpenEA 15K benchmark DBP1.0 are presented in Table 12. For this entity alignment experiment, training protocols of JMAC and baselines are the same as described in Sections 4.3 and A.1. Here, the test set results are reported for the model checkpoint which obtains the highest MRR on the validation set. Table 13 reports obtained alignment results on the test sets, where our JMAC performs better than the baselines in both the "w/ SI" and "w/o SI" categories, obtaining new state-of-the-art performances.

Method	**EL - EN**	**EL - ES**	**EL - FR**	**EL - JA**	**EN - FR**	**ES - EN**	**ES - FR**	**JA - EN**	**JA - ES**	**JA - FR**	**Overall**
						w/o SI					
AlignKGC	-	-	-	-	-	-	-	-	-	-	50.2
MTransE	24.2	34.9	30.0	38.6	20.6	23.6	31.6	19.2	26.7	41.6	28.2
AliNet	51.5	**73.5**	53.7	69.1	51.7	67.0	70.1	48.7	56.2	**75.1**	61.3
JMAC	**61.1**	72.8	**64.3**	**69.3**	**59.0**	**67.2**	**72.9**	**54.2**	**59.6**	64.7	**63.8**
						w/ SI					
AlignKGC	-	-	-	-	-	-	-	-	-	-	84.8
SS-AGA	16.2	15.9	15.0	2.7	70.4	79.8	76.4	6.8	5.6	4.8	34.1
PSR	77.1	79.5	74.0	75.2	76.9	86.3	86.8	68.1	66.3	80.1	77.2
RDGCN	93.0	88.6	88.0	84.9	89.6	**94.0**	88.7	91.1	84.0	89.0	89.3
JMAC	**93.1**	**92.8**	**92.4**	**93.8**	**94.5**	93.3	**94.5**	**94.7**	**91.9**	**93.1**	**93.4**

Table 6: MKGA Hits@1 results.

Method	**EL - EN**	**EL - ES**	**EL - FR**	**EL - JA**	**EN - FR**	**ES - EN**	**ES - FR**	**JA - EN**	**JA - ES**	**JA - FR**	**Overall**
						w/o SI					
AlignKGC	-	-	-	-	-	-	-	-	-	-	65.4
MTransE	43.2	54.1	45.3	55.4	33.4	39.4	48.0	32.9	42.1	58.8	44.0
AliNet	66.2	**82.7**	67.4	**79.6**	65.1	77.0	82.4	62.2	68.3	**84.3**	73.2
JMAC	**68.8**	80.6	**73.0**	79.4	**71.2**	**78.7**	**85.6**	**67.2**	**74.1**	80.3	**75.8**
						w/ SI					
AlignKGC	-	-	-	-	-	-	-	-	-	-	91.9
SS-AGA	23.3	24.2	23.1	5.1	76.1	86.1	81.4	13.2	11.8	10.6	40.1
PSR	87.9	91.3	86.4	88.3	90.3	92.3	92.2	86.0	83.0	86.2	88.4
RDGCN	97.3	95.7	94.9	91.1	95.3	**97.8**	94.6	95.1	91.3	94.9	94.9
JMAC	**97.5**	**97.8**	**96.8**	**97.6**	**97.8**	97.7	**97.8**	**98.2**	**95.8**	**97.6**	**97.5**

Table 7: MKGA Hits@10 results.

Method	**EL - EN**	**EL - ES**	**EL - FR**	**EL - JA**	**EN - FR**	**ES - EN**	**ES - FR**	**JA - EN**	**JA - ES**	**JA - FR**	**Overall**
						w/o SI					
MTransE	33.5	44.2	38.0	47.0	27.4	31.7	39.6	26.7	34.4	49.9	36.2
AliNet	58.4	**77.8**	60.2	**74.1**	57.9	71.7	75.7	54.9	61.8	**79.4**	60.8
JMAC	**63.6**	74.8	**67.0**	73.7	**65.1**	**72.8**	**80.4**	**62.4**	**69.1**	74.6	**70.3**
						w/ SI					
SS-AGA	20.2	20.5	19.8	4.6	73.2	82.6	78.8	10.6	9.2	8.3	37.4
PSR	81.1	83.8	78.1	79.9	81.6	88.6	88.9	74.3	72.3	81.9	81.2
RDGCN	95.1	91.8	91.3	87.9	92.2	**95.7**	91.4	93.1	87.4	91.7	91.9
JMAC	**95.3**	**95.0**	**94.6**	**95.2**	**94.6**	**95.7**	**96.0**	**96.3**	**93.3**	**95.3**	**95.1**

Table 8: MKGA MRR results. Singh et al. (2021) do not report the MRR results for AlignKGC.

Variants	**EL-EN**	**EL-ES**	**EL-FR**	**EL-JA**	**EN-FR**	**ES-EN**	**ES-FR**	**JA-EN**	**JA-ES**	**JA-FR**	**Overall**
JMAC w/ SI	**93.1**	**92.8**	**92.4**	**93.8**	**94.5**	**93.3**	**94.5**	**94.7**	**91.9**	**93.1**	**93.4**
(i) w/o RA-GNN	89.0	89.4	85.1	89.3	91.4	89.6	90.7	90.4	87.3	88.2	89.3
(ii) w/ 1-GNN	60.3	72.1	63.6	66.8	59.1	67.1	71.0	52.3	56.8	63.8	62.4
(iii) w/o SIR	90.1	90.4	87.8	91.7	92.9	91.3	92.6	92.5	89.9	90.8	91.3
(iv) w/o EnTr	92.6	92.1	88.9	93.5	94.0	93.1	94.1	93.9	91.2	92.9	92.9
(v) w/o Comple.	90.1	90.4	87.8	91.7	92.9	91.3	92.6	92.5	89.9	90.8	91.3

Table 9: Ablation Hits@1 results for the MKGA task.

Variants	**EL-EN**	**EL-ES**	**EL-FR**	**EL-JA**	**EN-FR**	**ES-EN**	**ES-FR**	**JA-EN**	**JA-ES**	**JA-FR**	**Overall**
JMAC w/ SI	**97.5**	**97.8**	**96.8**	**97.6**	**97.8**	**97.7**	**97.8**	**98.2**	**95.8**	**97.6**	**97.5**
(i) w/o RA-GNN	91.3	92.1	91.6	93.3	92.7	93.4	92.5	93.1	90.8	91.3	92.2
(ii) w/ 1-GNN	63.3	76.4	65.2	69.3	64.3	71.1	73.3	55.2	57.8	65.3	65.4
(iii) w/o SIR	94.1	95.3	93.6	94.1	95.2	95.0	94.2	95.1	92.3	94.4	94.3
(iv) w/o EnTr	95.6	95.9	94.9	95.0	95.8	96.1	95.8	96.2	94.2	96.4	95.6
(v) w/o Comple.	94.1	95.3	93.6	94.1	95.2	95.0	94.2	95.1	92.3	94.4	94.3

Table 10: Ablation Hits@10 results for the MKGA task.

Variants	EL-EN	EL-ES	EL-FR	EL-JA	EN-FR	ES-EN	ES-FR	JA-EN	JA-ES	JA-FR	Overall
JMAC w/ SI	**95.3**	**95.0**	**94.6**	**95.2**	**94.6**	**95.7**	**96.0**	**96.3**	**93.3**	**95.3**	**95.1**
(i) w/o RA-GNN	90.1	91.1	90.3	92.4	91.6	91.3	91.4	91.7	89.7	89.3	90.9
(ii) w/ 1-GNN	61.4	73.7	64.6	67.3	60.1	68.4	72.6	53.7	57.6	64.4	64.0
(iii) w/o SIR	93.2	93.1	91.4	91.6	92.4	93.6	92.7	93.5	90.3	93.6	92.6
(iv) w/o EnTr	95.0	94.6	93.2	93.5	94.3	95.5	94.8	95.3	92.3	95.2	94.5
(v) w/o Comple.	93.2	93.1	91.4	91.6	92.4	93.6	92.7	93.5	90.3	93.6	92.6

Table 11: Ablation MRR results for the MKGA task.

KG Pairs	KGs	V1			V2		
		#Entity	#Relation	#Triple	#Entity	#Relation	#Triple
EN-FR-15K	EN	15,000	267	47,334	15,000	193	96,318
	FR	15,000	210	40,864	15,000	166	80,112
EN-DE-15K	EN	15,000	215	47,676	15,000	169	84,867
	DE	15,000	131	50,419	15,000	96	92,632

Table 12: Statistics of bilingual KG pairs from the OpenEA 15K benchmark DBP1.0.

KG Pairs	Metric	w/o SI			w/ SI		
		MTransE	AliNet	JMAC	RDGCN	PSR	JMAC
EN-FR-15K V1	Hits@1	24.7	38.8	**58.8**	75.4	76.5	**90.0**
	Hits@10	56.3	82.9	**83.2**	88.1	93.2	**98.0**
	MRR	35.2	48.5	**67.4**	80.1	82.3	**93.0**
EN-FR-15K V2	Hits@1	24.1	58.1	**70.9**	84.7	92.5	**97.1**
	Hits@10	24.0	87.8	**89.0**	93.4	98.4	**99.6**
	MRR	33.7	69.2	**77.7**	88.0	94.3	**98.1**
EN-DE-15K V1	Hits@1	30.8	61.0	**73.2**	83.0	88.2	**94.3**
	Hits@10	61.1	83.1	**90.9**	91.4	95.5	**99.0**
	MRR	41.0	68.2	**79.5**	85.6	91.4	**96.1**
EN-DE-15K V2	Hits@1	19.4	81.5	**89.8**	83.4	96.5	**97.8**
	Hits@10	43.2	93.0	**97.1**	93.6	99.1	**99.5**
	MRR	27.4	85.6	**92.5**	86.1	97.7	**98.6**

Table 13: DBP1.0 test set results. We report our results for AliNet and PSR using their publicly released implementations. Results for MTransE and RDGCN are taken from Sun et al. (2020c). Here, RDGCN is the best performing model among 12 different models experimented by Sun et al. (2020c).