Knowledge Graph Completion with Joint Relation and Entity Alignment

BTP Phase 1 Report

by

Shubham Lohiya
(Roll No. 18D100020)

Supervisors:
Prof. Soumen Chakrabarti
Prof. Asim Tewari

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

2021

Contents

Chapter 1

Introduction

1.1 Background

Humans are inherently good at storing, reasoning and interpreting knowledge, and this enables us to learn to perform tasks from historically accumulated knowledge. Today there are billions of documents on the web. For a very long time, it has been the aim to computer science and AI to formulate a way for machines to interpret this *digital* knowledge through knowledge representation and reasoning. Recently, Knowledge Graphs have drawn a lot of research attention as form of structured fact representation. Figure 1.1 is an example of a small knowledge graph taken from [4].

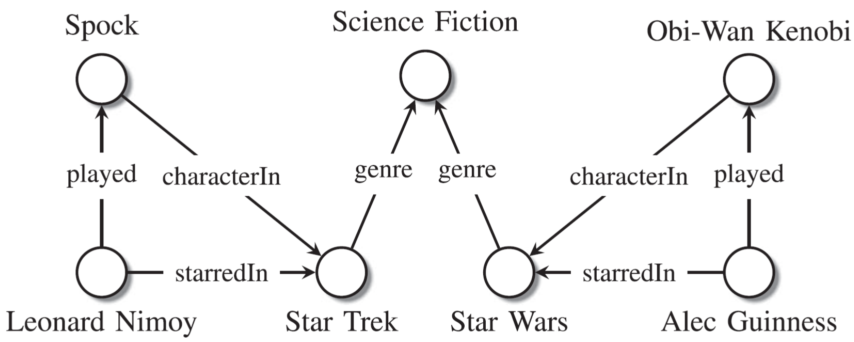

Figure 1.1: Sample knowledge graph. Nodes represent entities, edge labels represent types of relations, and edges represent existing relationships.

Knowledge graphs model information in the form of entities and relationships between them. A common form of fact representation in knowledge graphs is in the form of binary relationships, in particular `(subject, predicate, object)` or SPO triples, where the subject and the object are entities and the predicate is the relation between them. Entities can be real-world objects or abstract concepts. Another way to represent such fact triples is `(h, r, t)` which indicates a relation `r` between head entity `h` and tail entity `t`. In this report, the notation

‘e’ points to an entity e, and ‘$\mathbf{e}$’ refers to the feature vector representation or embedding of e. Similar notation follows for relations.

Some examples of real-world knowledge graphs include Freebase [2], DBPedia [1], Yago [7] and WikiData [12]. Knowledge graphs are extensively used on tasks like search engines, chat-bots, and recommendation systems. Biological Knowledge Graphs are also being researched and have applications like understanding molecular biology and drug discovery. However, knowledge graphs are generally far from exhaustive, and have especially sparse representation in low-resource languages. This has prompted research into techniques for Knowledge Graph Completion for predicting missing facts, and Knowledge Graph Alignments for integrating information from various graphs.

1.2 Report outline

The subject matter of the report is presented in the following 6 chapters,

- ✓ Chapter-1 provides an overview Knowledge Graphs (KGs) as a way of representing knowledge in the form of fact triples and discusses applications. It also introduces some problems with KGs that serve as the subject of recent research efforts. The outline of the report is also mentioned in this chapter.
- ✓ Chapter-2 describes all the new developments of methodologies for KG completion and alignment in separate subsections.
- ✓ Chapter 3 discusses the datasets and benchmarks for various knowledge graph research problems. It also presents important statistics and inferences gained from analyses of these datasets.
- ✓ Chapter-4 presents the existing research on the AlignKGC Framework in a concise manner, and introduces the work done in this project that supplements the published work.
- ✓ Chapter-5 explains the experiments conducted and highlights the new findings.
- ✓ Chapter-6 concludes the report by summarizing the research conducted and obtained results. Future work as a continuation of this research work is also proposed.

Chapter 2

Literature Review

2.1 KG completion

Knowledge Graph Completion (KGC) is the task of inferring missing facts based on existing data in a knowledge graph. Described below are some recent notable approaches for KGC.

2.1.1 ComplEx

ComplEx [11] calculates the confidence score of a candidate triple (s,r,o) using a Hermitian dot product on complex vector representations.

$$Pr(s,r,o) = \sigma(\Re(\langle \mathbf{s}, \mathbf{r}, \bar{\mathbf{o}} \rangle)) = \sigma(\Re(\sum_{k=1}^{K} r_k s_k \bar{o}_k)) \tag{2.1}$$

where $\mathbf{s}$, $\mathbf{r}$, and $\mathbf{o} \in \mathbb{C}^K$, $\Re(\cdot)$ takes the real part of a complex number, and $\bar{\cdot}$ denotes complex conjugate. The scoring function is quite similar to DistMult [14] which used a multi-linear dot product of real vectors and was unable to model anti-symmetric relations. ComplEx also has the same time and space complexity as TransE [3], which was unable to model symmetric relations. ComplEx is can handle both symmetric and anti-symmetric relations.

2.1.2 RotatE

RotatE [9] is an approach for knowledge graph embedding which defines each relation as a rotation from the head entity to the tail entity in the complex vector space. It improves over ComplEx, and is capable of modelling composition relations unlike ComplEx. RotatE uses a distance based scoring function which is defined as:

$$\phi(s,r,o) = -\left\|(\mathbf{s} \circ \mathbf{r} - \mathbf{o})\right\|^2 \tag{2.2}$$

where $\circ$ denotes the Hadamard Product.

RotatE also introduces a new approach to negative sampling during optimization. It shows that self-adversarial negative sampling method which samples negative triples according to the current embedding model is more effective than uniform negative sampling.

2.2 Entity and relation alignment

Entity Alignment is the task of finding entities in two KGs that refer to the same real-world object. It plays a vital role in automatically integrating multiple knowledge bases. Relations Alignment similarly aims to align relations in two KGs that refer to the same underlying relation. Described below are some recent approaches for the entity and relation alignment tasks.

2.2.1 BERT-INT

BERT-based Interaction Model For Knowledge Graph Alignment [10] is currently one of the best performing models for entity alignment. BERT-INT posits that due to heterogenity of different knowledge graphs and low available seed alignment information, alignment methods that primarily rely on graph structures to incorporate side information (such as name, description and attributes) result in noisy and inefficient results. It aims to leverage only side information and introduces an interaction module to capture fine-grained matches between neighbors and attributes. The BERT-INT architecture (refer Figure 2.1) consists of two modules:

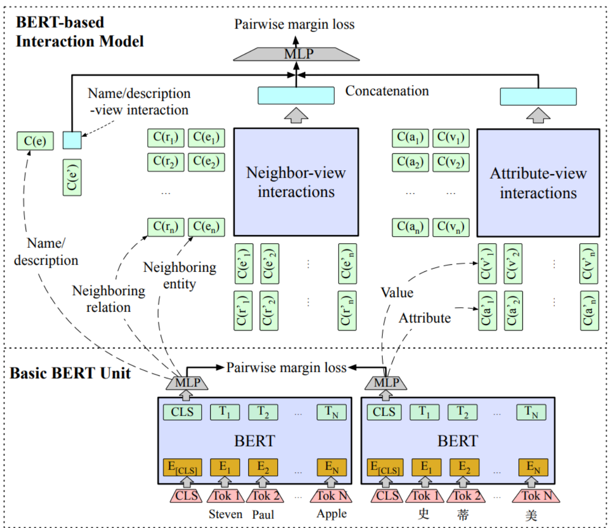

Figure 2.1: The BERT-INT framework for entity alignment.

1. **Basic BERT Unit**

This module uses pre-trained multilingual BERT[1] embeddings to calculate closest matches. Training data is of the form $\mathscr{D} = (e, e'^{+}, e'^{-})$. For each entity e in the dataset, the basic BERT unit takes its name/description as input to obtain $C(e) = MLP(CLS(e))$. The BERT model is fine-tuned with a pairwise margin loss seen in Equation 2.3.

$$\mathscr{L} = \sum_{(e, e'^{+}, e'^{-}) \in \mathscr{D}} max\{0, g(e, e'^{+}) - g(e, e'^{-}) + m\} \tag{2.3}$$

where m is the margin enforced between positive and negative pairs, and $g(e, e')$ is the l_1 distance capturing similarity between $C(e)$ and $C(e')$.

2. **BERT-based Interaction Model**

   The interaction model is based on BERT and consists of the following:

   (a) **Name/description-view interaction**: The cosine similarity between $C(e)$ and $C(e')$ is calculated as the name/description-view interaction.

   (b) **Neighbor-view Interactions**: The basic BERT unit is applied to obtain $C(e_i)_{i=1}^{|\mathscr{N}(e)|}$ and $C(e'_i)_{i=1}^{|\mathscr{N}(e')|}$ for e and e''s neighboring entities. Then, a similarity matrix is computed between the two embedding sets and a dual aggregation function is applied to extract the similarity features from the matrix. The matrix $\mathbf{S}$ represents the neighbor interaction based on cosine similarity. A a row-based similarity embedding $\phi^r(\mathscr{N}(e), \mathscr{N}(e'))$ is calculated as described in equation 2.4.

$$\begin{aligned}
s_i^{max} &= \max_{j=0}^{n}\{s_{i0}, \cdots, s_{ij}, \cdots, s_{in}\} \\
K_l(s_i^{max}) &= \exp\left[-\frac{(s_i^{max} - \mu_l)^2}{2\sigma_l^2}\right] \\
\mathbf{K}^r(\mathbf{S}_i) &= [K_1(s_i^{max}), \cdots, K_l(s_i^{max}), \cdots, K_L(s_i^{max})] \\
\phi^r(\mathscr{N}(e), \mathscr{N}(e')) &= \frac{1}{|\mathscr{N}(e)|}\sum_{i=1}^{|\mathscr{N}(e)|} \log \mathbf{K}^r(\mathbf{S}_i)
\end{aligned} \tag{2.4}$$

   A a column-based similarity embedding $\phi^c(\mathscr{N}(e), \mathscr{N}(e'))$ is computed in a similar manner. The final similarity embedding is computed by concatenating them.

$$\phi(\mathscr{N}(e), \mathscr{N}(e')) = \phi^r(\mathscr{N}(e), \mathscr{N}(e')) \oplus \phi^c(\mathscr{N}(e), \mathscr{N}(e')) \tag{2.5}$$

[1] https://github.com/google-research/bert

(c) **Neighboring Relation Mask Matrix**: The similarity matrix **S** is modified using matrix **M** which calculated from relation embeddings $C(r)$. $C(r)$ is computed by concatenating averaged embeddings of all head and tail entities associated with relation r. Finally **S** is recalculated as follows:

$$S_{ij} = S_{ij} \otimes M_{ij} \tag{2.6}$$

(d) **Interactions between Multi-hop Neighbors**: In the similarity matrix instead of taking 1-hop neighbours, m-hop neighbours are used.

(e) **Attribute-view Interactions**: $\phi(\mathscr{A}(e), \mathscr{A}(e'))$ are computed in the same manner as neighborhood interaction

The final interaction embedding is computed as equation 2.7 and the similarity score is defined in equation 2.8.

$$\phi(e,e') = [\phi(\mathscr{N}(e), \mathscr{N}(e')) \oplus \phi(\mathscr{A}(e), \mathscr{A}(e')) \oplus cos(C(e), C(e'))] \tag{2.7}$$

$$g(e,e') = MLP(\phi(e,e')) \tag{2.8}$$

2.2.2 RNM

Most methods for EA aggregate information from neighboring nodes but they ignore the relations between entities, which are also important for neighborhood matching. The Relation-aware Neighborhood Matching model for entity alignment (RNM) [16] pays attention to the positive interactions between the entity alignment and relation alignment. It presents an iterative framework designed to leverage the positive interactions between the EA and RA in a semi-supervised manner. The components of the proposed architecture (refer Figure 2.2) are described below:

1. **Preliminaries**: The objective is to align entities given two heterogeneous KGs, $G_1 = (E_1, R_1, T_1)$ and $G_2 = (E_1, R_1, T_1)$. A set $\mathbb{L} = \{(e_1, e_2) | e_1 \in E_1, e_2 \in E_2, e_1 \ equals \ to \ e_2\}$ is provided to the model as seed alignment.

2. **Embedding Learning for Entity and Relation**:
   Entity Embedding: GCNs are utilized to embed all entities of the two KGs into the same latent space. Pre-trained word embeddings are used to initialize the entity representations.

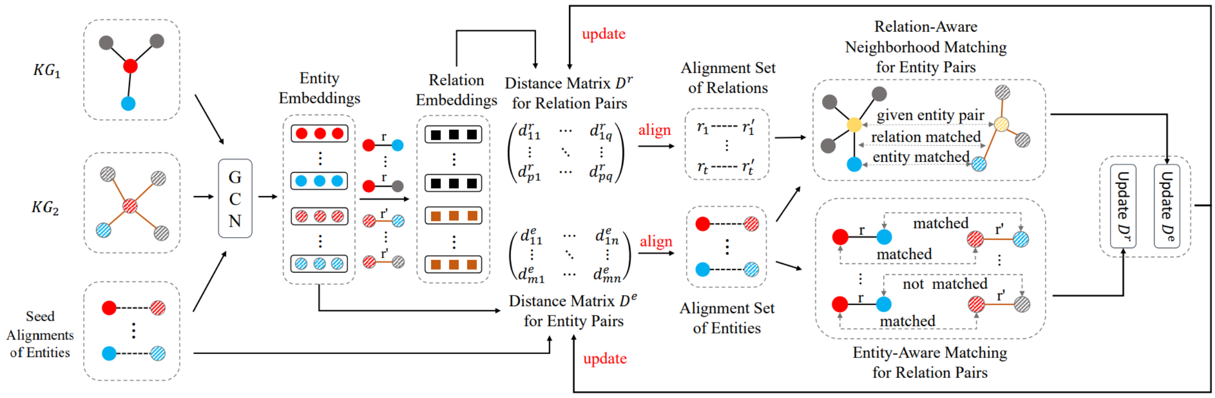

Figure 2.2: Overall architecture of the RNM model for entity and relation alignment.

The outputs of the GCN define the final entity representations as $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_\mathbf{1}, \tilde{\mathbf{x}}_\mathbf{2}, \cdots \tilde{\mathbf{x}}_\mathbf{n} | \tilde{\mathbf{x}}_\mathbf{i} \in \mathbb{R}^{\tilde{d}}\}$. For an entity pair (e_i, e'_j), the distance between them is defined as

$$d(e_i, e'_j) = ||\tilde{\mathbf{x}}_{\mathbf{e_i}} - \tilde{\mathbf{x}}_{\mathbf{e'_j}}||_1 \tag{2.9}$$

The embeddings are trained by optimizing the margin-based loss given in equation 2.10

$$L_E = \sum_{(p,q)\in\mathbb{L}} \sum_{(p',q')\in\mathbb{L}'} \max\left\{0, d(p,q) - d\left(p',q'\right) + \gamma\right\} \tag{2.10}$$

where $\mathbb{L}'$ is a set of negative alignments created using nearest neighbor sampling, and $\gamma > 0$ denotes the margin.

Relation Embedding: The relation embedding is computed as follows

$$\boldsymbol{r} = \text{concat}\left[\boldsymbol{g}_r^h, \boldsymbol{g}_r^t\right] \tag{2.11}$$

where $\boldsymbol{r} \in \mathbb{R}^{2\tilde{d}}$, and $\boldsymbol{g}_r^h$ and $\boldsymbol{g}_r^t$ denote average embeddings of all distinct head and tail entities for r.

To further explore the translational information for relations based on triples, a regularizer is defined as

$$\Omega_R = \sum_{(h,r,t)\in T_1 \cup T_2} \|\boldsymbol{h} + \boldsymbol{W}_R \boldsymbol{r} - \boldsymbol{t}\|_1 \tag{2.12}$$

where $\boldsymbol{W}_R \in \mathbb{R}^{\tilde{d}\times 2\tilde{d}}$ denotes the transformation matrix from the latent relation space to the latent entity space that has to be learned.

The joint learning objective is defined in equation 2.13

$$L = L_E + \lambda \cdot \Omega_R \tag{2.13}$$

3. **Relation-Aware Neighborhood Matching**: GCNs aim to aggregate information from neighboring nodes but may also bring some additional noise from neighbors. To reduce the impact of these noise, Relation-Aware neighborhood matching is done and the entity distance matrix is iteratively updated. If two entities from different KGs are equivalent, then with an equivalent relation, the alignment probability of two pointing tail entities is inferred to be 1 for a 1-to-1 relation while 1-to-N relation can only show the probability of $1/N$.

   For neighborhood matching with respect to e_i and e'_j, all the entity pairs and the connected relation pairs in $C^e_{ij} = \{(n_1, n_2), (r_1, r_2) \mid n_1 \in \mathcal{N}_{e_i}, n_2 \in \mathcal{N}_{e'_j}, (e_i, r_1, n_1) \in T_1, (e'_j, r_2, n_2) \in T_2\}$ are to be compared. Then the focus is on the matched neighbors with matched relations which are vital for entity alignment, so we define the matched set M^e_{ij} as the subset of C^e_{ij}, in which the elements satisfy $(n_1, n_2) \in \mathbb{L}_e$ and $(r_1, r_2) \in \mathbb{L}_r$, where $\mathbb{L}_e$ denotes the alignment set of entities and $\mathbb{L}_r$ denotes the alignment set of relations. These sets are obtained every iteration using the seed alignments and additional discovered alignments by thresholding the distance matrices.

   For each matched case in M^e_{ij}, the alignment probability is computed as follows,

$$\begin{aligned} P(r_1, r_2, n_1, n_2) &= P(r_1, n_1) \cdot P(r_2, n_2) \\ \text{where} \quad P(r_1, n_1) &= \frac{1}{|\{e \mid (e, r_1, n_1) \in T_1\}|} \\ \text{and} \quad P(r_2, n_2) &= \frac{1}{|\{e \mid (e, r_2, n_2) \in T_2\}|} \end{aligned} \tag{2.14}$$

   and the distance between two entities is updated as follows,

$$d^e_{ij} = ||\tilde{\boldsymbol{x}}_{e_i} - \tilde{\boldsymbol{x}}_{e'_j}||_1 - \lambda_e \cdot \frac{\sum_{M^e_{ij}} P(r_1, r_2, n_1, n_2)}{|\mathcal{N}_{e_i}| + |\mathcal{N}_{e'_j}|} \tag{2.15}$$

   where λ_e is a hyper-parameter to control the trade-off between the embedding distance and the matching score. Greater matching score indicates the higher probability of alignment for the candidate entity pair.

4. **Entity-Aware Relation Matching**:

   For two relations from different KGs, it is assumed that high number of (head, tail) alignments of the corresponding triples accross the KGs indicates higher likelihood that the two relations are equivalent. $S_r = \{(h, t) \mid (h, r, t) \in T\}$ is defined as the set of its related entity pairs. Given a candidate relation pair (r_i, r'_j), all entity pairs in $C^r_{ij} = \{(h_1,h_2),(t_1,t_2) \mid (h_1,t_1) \in S_{r_i},(h_2,t_2) \in S_{r'_j}\}$ are to be compared and the matching set M^r_{ij} is defined as the subset of C^r_{ij} where elements meet the conditions of $(h_1,h_2) \in \mathbb{L}_e$ and $(t_1,t_2) \in \mathbb{L}_e$. Then, the distance between the two relations is updated as follows,

$$d^r_{ij} = \|\boldsymbol{r}_i - \boldsymbol{r}'_j\|_1 - \lambda_r \cdot \frac{|M^r_{ij}|}{|S_{r_i}| + |S_{r'_j}|} \tag{2.16}$$

where λ_r is the trade-off coefficient.

2.3 Text and attribute features

Text descriptions of entities and attributes can provide useful context signals for KG completion and alignment tasks. Described next are some techniques to incorporate this information for such tasks.

2.3.1 mBERT embeddings

BERT is a language representation model that can be used to obtain general purpose contextual word embeddings. BERT and its variants have been highly successful on various downstream NLP tasks, achieving state of the art results. Unlike fixed word embeddings like Word2Vec and GloVe, BERT embeddings capture context from surrounding text. mBERT is a Multilingual version of BERT.

BERT takes as input, N wordpiece tokens: $(x_1,\ldots,x_N)$. Following that, L layers of D-dimensional contextual representations $\mathbf{H}_i \in \mathbb{R}^{N\times D}$ are calculated by successive application of non-linear functions $\mathbf{H}_i = \mathrm{F}_i(\mathbf{H}_{i-1})$. The non linear function, F_i, is a multi-headed self-attention layer followed by a position-wise multi-layer perceptron (MLP).

$$\mathrm{F}_i(\mathbf{H}_{i-1}) = \mathrm{TransformerBlock}(\mathbf{H}_{i-1}) = \mathrm{MLP}(\mathrm{MultiHeadAttn}(\mathbf{H}_{i-1},\mathbf{H}_{i-1},\mathbf{H}_{i-1})) \tag{2.17}$$

BERT-INT [10] demonstrated the benefits of using text embeddings for entity alignment, by achieving state of the art results using just this side information instead of graph structures. It used mBERT embeddings to perform name/description-view interaction, neighbor-view interaction with relation masking, and attribute-view interaction as described in section 2.2.1. KG-BERT [15] has also demonstrated success in knowledge graph completion.

2.3.2 GloVe

GloVe [5] is an unsupervised learning algorithm for obtaining vector representations for words. As previously shown, the RNM model for entity alignment uses GloVe representations to initialise their GCN embeddings. Entity names from different languages are first translated to english to obtain the GloVe representations. As described in Chapter 5, experiments with RNM on DBP-5L and DBP15K datasets using both mBERT and GloVe initialisations yield some interesting insights about the benefit of using these vectors.

2.3.3 Other methods to generate entity representations

This sections describes a few methods to generate set representations of entities as mentioned in [13]. The paper used these set representations to perform Locality-Sensitive Hashing (LSH) for obtaining candidate pairs for entity alignment.

- **N-grams of Names**: If entity names are available and in the same language, this method generates a set of character-level n-grams of entities' names as the set-representations of entities.
- **N-grams of Attributes**: This method treats attribute values of an entity as text strings, and generates character-level n-grams of all the attribute values for each entity. All the n-grams are then merged into a set as the representation of the entity.
- **Seeding alignments**: If seeding alignments between two KGs are available, a set of aligned entities in an entity's neighborhood will be taken as the set-representation.

Chapter 3

Datasets

3.1 DBP-5L

DBP-5L contains KGs for 5 languages - Greek, Japanese, Spanish, French and English. It was sampled from the DBPedia knowledge base. This dataset is applicable to all three tasks – KGC, EA and RA

3.1.1 Data Statistics

Salient statistics of the DBP-5L dataset are mentioned in table 3.1

Language	Greek	Japanese	Spanish	French	English
#Entity	5,231	11,805	12,382	13,176	13,996
#Relation	111	128	144	178	831
#Triples	13,839	28,774	54,066	49,015	80,167

Table 3.1: Data distribution statistics for DBP5L

For KGC, the split of fact triples between the training, the validation and the test sets is approximately in the ratio of *60:30:10* for all 5 languages. Similar splits have been followed for sampling while converted other datasets like DBP15K to allow them to be used for testing KGC.

3.2 DBP15K

DBP15k is the most popular benchmark for Entity Alignment, and was also sampled from the DBPedia knowledge base. The data set has three multilingual KG pairs: *ZH_EN* (Chinese-English), *JA_EN* (Japanese-English), and *FR_EN* (French English).

3.2.1 Data Statistics

Salient statistics of the datasets are mentioned in table 3.2

Datasets		Entities	Relationships	Attributes	Rel. Triples	Attr. triples
$DBP15K_{ZH-EN}$	Chinese	66,469	2,830	8,113	153,929	379,684
	English	98,125	2,317	7,173	237,674	567,755
$DBP15K_{JA-EN}$	Japanese	65,744	2,043	5,882	164,373	354,619
	English	95,680	2,096	6,066	233,319	497,230
$DBP15K_{FR-EN}$	French	66,858	1,379	4,547	192,191	528,665
	English	105,889	2,209	6,422	278,590	576,543

Table 3.2: Data distribution statistics for DBP15k

3.2.2 EA Results

Results of various approaches to entity alignment on the DBP15K benchmark are shown in table 3.3

Models	ZH-EN			JA-EN			FR-EN		
	Hits@1	Hits@10	MRR	Hits@1	Hits@ 10	MRR	Hits@1	Hits@10	MRR
MTransE (Chen et al. 2017)	30.8	61.4	0.364	27.9	57.5	0.349	24.4	55.6	0.335
IPTransE (Zhu et al. 2017)	40.6	73.5	0.516	36.7	69.3	0.474	33.3	68.5	0.451
BootEA (Sun et al. 2018)	62.9	84.8	0.703	62.2	85.4	0.701	65.3	87.4	0.731
AKE (Lin et al. 2019)	32.5	70.3	0.449	25.9	66.3	0.390	28.7	68.1	0.416
SEA (Pei et al. 2019)	42.4	79.6	0.548	38.5	78.3	0.518	40.0	79.7	0.533
GCN-Align (Wang et al. 2018)	41.3	74.4	0.549	39.9	74.5	0.546	37.3	74.5	0.532
KECG (Li et al. 2019)	47.8	83.5	0.598	49.0	84.4	0.610	48.6	85.1	0.610
MuGNN (Cao et al. 2019a)	49.4	84.4	0.611	50.1	85.7	0.621	49.5	87.0	0.621
NAEA (Zhu et al. 2019)	65.0	86.7	0.720	64.1	87.3	0.718	67.3	89.4	0.752
AliNet (Sun et al. 2020)	53.9	82.6	0.628	54.9	83.1	0.645	55.2	85.2	0.657
GMNN (Xu et al. 2019)	67.9	78.5	0.694	74.0	87.2	0.789	89.4	95.2	0.913
RDGCN (Wu et al. 2019a)	70.8	84.6	0.746	76.7	89.5	0.812	88.6	95.7	0.911
HGCN (Wu et al. 2019b)	72.0	85.7	0.768	76.6	89.7	0.813	89.2	96.1	0.917
NMN (Wu et al. 2020)	73.3	86.9	0.781	78.5	91.2	0.827	90.2	96.7	0.924
BERT-INT	**96.8**	**99**	**0.977**	**96.4**	**99.1**	**0.975**	**99.2**	**99.8**	**0.995**
RNM	*84.0*	*91.9*	*0.870*	*87.2*	*94.4*	*0.899*	*93.8*	*98.1*	*0.954*

Table 3.3: EA results on DBP15K. These have been taken from the RNM paper and BERT-INT results have been taken from the BERT-INT paper

This clearly shows that RNM and BERT-INT hugely outperform the previous methods for entity alignment. RNM makes use of both word embeddings and graph structures, whereas BERT-INT relies only on side information for entity alignment as discussed in section 2.2.1. Hence, these two are our primary algorithms of interest for comparision with AlignKGC

3.3 DBP2.0

Since KGs possess different sets of entities, there could be entities that cannot find alignment across them, leading to the problem of *dangling entities*. As previous datasets do not contain dangling entities, Sun et al. [8] design a new dataset that can be used for both entity alignment and dangling entity detection. This Dataset was also sampled from DBPedia.

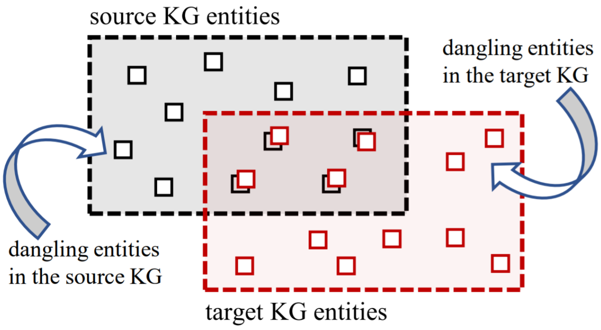

Figure 3.1: Illustration of entity alignment between two KGs with dangling cases. Paired red and black squares in the overlap region denote entity alignment while others are dangling entities without counterparts

3.3.1 Dataset Construction

A two-step dataset extraction is followed to ensure that the selected dangling entities are indeed without counterparts. First, two subgraphs are sampled for which all entities have alignments in the other subgraph. Then, a randomly selected disjoint set of entities are removed from the source and target graphs, to make their counterparts dangling.

3.3.2 Data Statistics

Salient statistics of the DBP2.0 dataset are mentioned in table 3.4

Datasets		# Entities	# Rel.	# Triples	# Align.
ZH-EN	ZH	84,996	3,706	286,067	33,183
	EN	118,996	3,402	586,868	
JA-EN	JA	100,860	3,243	347,204	39,770
	EN	139,304	3,396	668,341	
FR-EN	FR	221,327	2,841	802,678	123,952
	EN	278,411	4,598	1,287,231	

Table 3.4: Statistics of the DBP2.0 dataset

Chapter 4

AlignKGC

4.1 Introduction

The key objective of AlignKGC [6] is to recognize and exploit the synergy between Multilingual Knowledge Completion, Entity Alignment and Relation Alignment. High confidence fact predictions may add valuable information for alignment tasks, and vice versa. Findings from experiments with DBPedia with 5 languages indicate that AlignKGC achieves up to 17% absolute MRR improvements in mKGC compared to a strong completion model that combines known facts in all languages. It also outperforms mBERT on EA, demonstrating the value of joint training for these tasks. This project aims to extend the previously published research on AlignKGC.

4.2 Proposed Methods

As mentioned before, AlignKGC is a multi-task system that learns to optimize for KGC, EA and RA simultaneously. This section discusses the loss components of the joint optimization objective.

4.2.1 KGC Loss

The KGC component in AlignKGC is an extension of the near state-of-the-art ComplEx [11] which defines a triples score as

$$f(s,r,o) = \Re\left(\langle s,r,o^{\star}\rangle\right) \tag{4.1}$$

where $c^{\star}$ is complex conjugate, $\langle\cdots\rangle$ is a 3-way elementwise inner product and $\Re(\cdot)$ is the real part of a complex number. Using f, ComplEx defines

$$\begin{aligned}\Pr(o \mid s,r) &= e^{f(s,r,o)} / \sum_{o'} e^{f(s,r,o')} \\ \Pr(s \mid o,r) &= e^{f(s,r,o)} / \sum_{s'} e^{f(s',r,o)}\end{aligned} \tag{4.2}$$

and the log-likelihood KGC loss as

$$L_{\text{KGC}} = \sum_{(s,r,o)\in \text{KG}} -\log \Pr(o \mid s,r) - \log \Pr(s \mid o,r) \tag{4.3}$$

4.2.2 RA Loss

To formulate the RA Loss term, the Hard SO-signature of a relation is defined as $\text{SO}(r) = \{(s,o) : (s,r,o) \in T\}$. The SO-overlap between two relations $r_l, r_{l'}$ is $|\text{SO}(r_l) \cap \text{SO}(r_{l'})|$. Jaccard similarity can then be used to compute a symmetric belief that two relations in different languages are equivalent:

$$b_J(r_l \Leftrightarrow r_{l'}) = \frac{|\text{SO}(r_l) \cap \text{SO}(r_{l'})|}{|\text{SO}(r_l) \cup \text{SO}(r_{l'})|} \tag{4.4}$$

If $b_J(r_l \Leftrightarrow r_{l'})$ exceeds a threshold θ (tuned hyperparameter), $(r_l, r_{l'})$ are added to the set $\mathscr{A}_J$ of 'silver' alignments.

To handle asymmetric relations, the belief score is modified to $b_A(r_l \Leftrightarrow r_{l'})$ defined as

$$b_A(r_l \Leftrightarrow r_{l'}) = \frac{|\text{SO}(r_l) \cap \text{SO}(r_{l'})|}{max\{|\text{SO}(r_l)|, |\text{SO}(r_{l'})|\}} \tag{4.5}$$

These ideas are then extended to re-define the SO-signature using entity embeddings, which can be trained via gradient descent. The Soft SO-signature is defined as $\text{SO}(r) = \{(\mathbf{s},\mathbf{o}) : (s,r,o) \in T\}$, where each element is the concatenatin of the subject and object embedding vectors. Then the embeddings pairs in these sets can be compared by extending cosine-similarity:

$$\text{sim}((\mathbf{s},\mathbf{o}),(\mathbf{s}',\mathbf{o}') = \sigma(\cos(\mathbf{s},\mathbf{s}')) \cdot \sigma(\cos(\mathbf{o},\mathbf{o}')) \tag{4.6}$$

Then the Soft-SO Overlap as the continuous extension of $|\text{SO}(r_l) \cap \text{SO}(r_{l'})|$, denoted by $\text{SoftOv}(r_l, r_{l'})$ is defined as the value of the maximal matching on the weighted bipartite graph induced by $A_{r_l, r_{l'}}$, where

$$A_{r_l,r_{l'}}[i,j] = \text{sim}(\text{SO}(r_l)[i], \text{SO}(r_{l'})[j]) \tag{4.7}$$

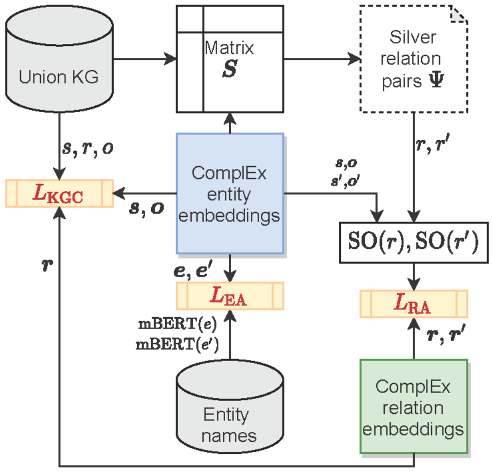

Figure 4.1: AlignKGC Architecture

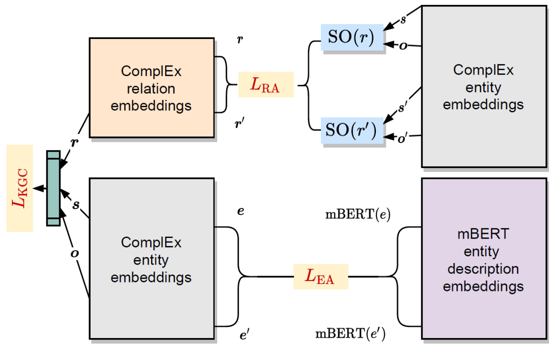

Figure 4.2: Loss components of joint objective

$b_{SA}(r_l \Leftrightarrow r_{l'})$ is defined by replacing terms in $b_A(r_l \Leftrightarrow r_{l'})$ with their 'soft' counterparts which leads to the soft asymmetric RA loss term for AlignKGC as shown below:

$$L_{\mathrm{RA-SA}} = \sum_{(r_l, r_{l'}) \in \mathscr{A}_{SA}} b_{SA}(r_l \Leftrightarrow r_{l'}) \|r_l - r_{l'}\|_1 \tag{4.8}$$

4.2.3 EA Loss

The loss component for entity alignment is

$$L_{\mathrm{EA}} = \sum_{e_l, e_{l'}} \cos(\mathrm{mbert}(e_l), \mathrm{mbert}(e_{l'})) \|e_l - e_{l'}\|_1 \tag{4.9}$$

where $\mathrm{mbert}(e_l)$ is the textual embedding of entity e mapped in the KG space using a MLP. The complete AlignKGC loss term is then defined as

$$L_{\mathrm{KGC}} + \alpha L_{\mathrm{reg}} + \beta L_{\mathrm{RA-SA}} + \gamma L_{\mathrm{EA}} \tag{4.10}$$

where L_{reg} is an L_2 regularization on embeddings and $\alpha, \beta, \gamma \geq 0$ are tuned hyperparameters.

4.3 Discussion

The main criticism that AlignKGC faced was that it lacked proper EA and RA baselines on DBP-5L. This work aims to rectify that by comparing notable alignment algorithms with AlignKGC on both DBP-5L and the popular DBP15K benchmark. It also aims to propose extensions to AlignKGC as mentioned in Chapter 6.

Chapter 5

Experiments

Since the RNM model is the current state-of-the-art among approaches to do entity and relation alignment that use graph structures, it has been chosen as the competitive baseline for comparison with AlignKGC on these tasks.

5.1 AlignKGC vs RNM on DBP-5L

DBP-5L was converted to a format similar to DBP15K, for input to RNM. The following modifications needed to be made:

- Initialization of GCN embedding was done using mBERT intead of GloVe for experiments with RNM on DBP-5L, to ensure fair comparison with AlignKGC
- A small fraction of the seed alignments from DBP-5L needed to be dropped, as only the facts exposed during training to AlignKGC were given as input to RNM for entity alignment. So the nodes in this subsampled KG that do not have an outgoing relation were dropped from the set of aligned entities before sampling the seed alignment for RNM, as it was requirement by the neighborhood matching algorithm.

Detailed results of RNM on DBP-5L are shown in table 5.1

	EA			RA	
	Hits@1	Hits@10	MRR	Hits@1	Hits@ 10
EL-EN	21.11	36.31	0.2644	25.36	55.07
EN-EL	20.66	32.47	0.2515	26.81	49.28
EL-ES	29.84	43.03	0.3472	46.4	71.2
ES-EL	29.33	40.91	0.3343	46.4	60
EL-FR	20.58	34.86	0.2566	35.92	59.22
FR-EL	21.38	35.04	0.2625	29.13	50.49
EL-JA	10.92	20.56	0.1443	31.09	59.66
JA-EL	12.65	22.02	0.1619	39.5	63.87
JA-EN	8.84	19.68	0.1275	38.21	65.85
EN-JA	8.2	17.1	0.1161	32.52	52.85
JA-ES	8.62	16.52	0.118	35.58	57.69
ES-JA	8.14	15.22	0.1089	32.69	52.88
JA-FR	13.16	20.51	0.1605	43.55	64.52
FR-JA	12.19	19.91	0.1506	37.1	61.29
ES-FR	73.36	83.56	0.7707	68.65	80
FR-ES	74.99	85.03	0.7868	67.03	80.54
ES-EN	77.7	85.51	0.8053	57.59	76.58
EN-ES	80.82	87.91	0.8347	55.7	75.32
EN-FR	75.15	82.52	0.7791	45.65	65.94
FR-EN	73.94	82.36	0.771	39.13	67.39

Table 5.1: Entity and Relation Alignment performance of RNM on DBP-5L

Comparison with AlignKGC: Performance of AlignKGC and RNM has been compared in table 5.2.

	HITS@1	
	AlignKGC	**RMN**
EL-EN	83.10	20.89
EL-ES	89.10	29.59
EL-FR	86.80	20.98
EL-JA	85.20	11.79
JA-EN	81.50	8.52
JA-ES	83.95	8.38
JA-FR	84.75	12.68
ES-FR	91.60	74.18
ES-EN	92.10	79.26
EN-FR	87.50	74.55
Average	**86.56**	34.08

Table 5.2: EA performance comparison on DBP-5L. Note that the number reported are mean of the left and right alignment performances

We observe that AlignKGC outperforms RMN for EA on DBP-5L. What's more interesting is that RMN is quite competitive for language pairs in which both languages have latin roots (colored green), but if even one language pair does not have a latin root, the performance drop is very drastic (colored orange). Thus suggests that there is insufficient representation of these languages in mBERT's dictionary. Thus the choice of mBERT embeddings for introducing text signals warrants further experimentation.

5.2 mBERT vs GloVe representations for Entity Alignment

Due to the observations mentioned in the previous section, experiments to compare mBERT with GloVE representations were conducted using the RNM model on the DBP15K dataset. Table 5.3 shows the results of these experiments. Note that for the changed initialisation, best hyperparameters were found using grid search.

	GloVe initialization (default)					**mBERT initialization**				
	EA			**RA**		**EA**			**RA**	
	Hits@1	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**	**Hits@1**	**Hits@10**	**MRR**	**Hits@1**	**Hits@10**
FR-EN	93.79	97.94	0.9537	49.53	62.74	84.88	91.99	0.8764	46.23	58.49
EN-FR	93.73	98.33	0.9549	50	61.32	86.64	93.63	0.8931	45.28	55.66
JA-EN	86.41	94.51	0.8945	75.24	86.01	47.85	55.4	0.5074	58.22	74.48
EN-JA	86.54	94.97	0.8973	72.97	83.93	48.45	57.78	0.5190	52.55	65.78
ZH-EN	84.67	91.84	0.8739	81.01	87.75	54.05	61.17	0.5683	65.17	79.55
EN-ZH	83.41	92.26	0.8675	80.11	86.85	53.25	61.85	0.5658	61.35	72.81

Table 5.3: Comparison of GloVe and mBert initialisation for RNM

We see that GloVe outperforms mBERT for initialisation of GCN embeddings, but the most important observation is that the drop in performance is a lot more for language pairs with at-least one language with a non-latin root. This further confirms the hypothesis that mBERT inadequately represents these languages in its dictionary.

Chapter 6

Conclusions and Next Steps

The AlignKGC framework jointly learns 3 important tasks for Multilingual Knowledge Graphs, namely Knowledge Graph Completion, Entity Alignment and Relation Alignment. The results presented in the previously published work, along with the new experiments presented in this report show great promise for this approach. But, these experiments have also exposed key problems with AlignKGC that warrant further examination.

6.1 Next Steps

- The choice of mBERT for AlignKGC needs further analysis. The translation + GloVe representation approach of RNM can be leveraged.
- The RA loss term for AlignKGC needs to be reformulated such that it can be included for every update instead of every few iterations.
- A new dataset which is more representative of real world data can be sampled from WikiData, which should incude:
  - Ambiguity seen in real world data
  - Multilinguality as well as heterogeneity
  - One to many relationships
  - *Dangling* entities as characteristic of DBP2.0
  - Text descriptions of entities attribute triples

  This dataset would aim to establish a new benchmark for all KG related tasks discussed in this report.
- Implementation of a GNN-based version of AlignKGC which reformulates the KG as a tripartite graph, which has entities in the left layer and relations in the right layer with fact triples forming the middle layer.

References

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[2] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[4] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

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

[6] Harkanwar Singh, Prachi Jain, Mausam, and Soumen Chakrabarti. Multilingual knowledge graph completion with joint relation and entity alignment, 2021.

[7] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[8] Zequn Sun, Muhao Chen, and Wei Hu. Knowing the no-match: Entity alignment with dangling cases. *arXiv preprint arXiv:2106.02248*, 2021.

[9] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

[10] Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. Bert-int: A bert-based interaction model for knowledge graph alignment. In *IJCAI*, pages 3174–3180, 2020.

[11] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

[12] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[13] Zhichun Wang, Jinjian Yang, and Xiaoju Ye. Knowledge graph alignment with entity-pair embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1672–1680, 2020.

[14] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

[15] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion, 2019.

[16] Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. Relation-aware neighborhood matching model for entity alignment, 2020.