Experimental Study on Knowledge Base Enrichment Algorithms

by

Fujia Zhou (978762)

Supervisor: Professor Rui Zhang
Project Type: Research Project
Subject Code: COMP90055
Credit Points: 25

School of Computing and Information Systems
Melbourne School of Engineering
THE UNIVERSITY OF MELBOURNE

November 2020

THE UNIVERSITY OF MELBOURNE

Abstract

School of Computing and Information Systems
Melbourne School of Engineering

Master of Information Technology

Experimental Study on Knowledge Base Enrichment Algorithms

by Fujia Zhou (978762)

In the past few years, knowledge bases have been widely used in plenty of AI-related research and applications, such as information extraction, question answering systems, and recommender systems. These knowledge bases are modelled as knowledge graphs (KGs) to store human knowledge, and they can be integrated by knowledge base enrichment algorithms for better knowledge fusion and inference. One typical way to consolidate knowledge and enhance the quality of knowledge graphs is through entity alignment, which aims to discover entities from different knowledge graphs that represent the same real-world object. While recent years have witnessed considerable progress in the development of entity alignment approaches, very few research has provided systematic comparisons and analysis of the existing approaches through experimental studies and empirical evaluations. Moreover, it has been overlooked by the current literature that existing benchmark datasets oversimplify the real-world challenges of KG entity alignment. With that said, this research project strives to investigate three state-of-the-art entity alignment approaches by summarising the related literature, reviewing the framework of each approach, deriving datasets to mirror the real-world challenges, carrying out a set of exploratory experiments, and suggesting potential research directions for future work.

Declaration of Authorship

I, Fujia Zhou, certify that:

- this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.
- where necessary I have received clearance for this research from the University's Ethics Committee and have submitted all required data to the School.
- the thesis is 7402 words in length (excluding text in images, table, bibliographies and appendices).

Signed: 周福佳

Date: 12 Nov 2020

Acknowledgements

I would like to express my sincere gratitude towards Professor Rui Zhang and Bayu Distiawan Trisedya from the School of Computing and Information Systems at the University of Melbourne for their supervision and contribution of ideas and directions to explore throughout my research project. This thesis would not have been possible without their guidance, support, and encouragement along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent years have witnessed the rapid development of knowledge bases and related applications. Many knowledge bases, such as Freebase [3], DBpedia [4], YAGO [5], and Wikidata [6], have been published during the past decades and are becoming essential sources of knowledge for AI-related research and applications, such as question answering systems [7] and recommender systems [8]. These knowledge bases are modelled as knowledge graphs (KGs) to store human knowledge as triples, which are generally in the form of $(subject\ entity, relationship, object\ entity)$ or $(entity, attribute, value)$.

In practice, KGs are constructed for various purposes and applications. As a result, a single KG is unlikely to have full coverage of different domains. In this case, KGs can be complementary to each other, and they can be integrated for better knowledge inferences. To enhance knowledge fusion, researchers have made considerable efforts throughout the years on developing entity alignment approaches, which become an essential type of knowledge base enrichment algorithms. Specifically, the goal of entity alignment is to identify entities in different KGs that represent the same real-world object.

Generally, existing entity alignment approaches assume that equivalent entities in different KGs tend to have similar neighbourhood structures, based on which they employ representation learning methods to project entities into low-dimensional vector spaces. In this way, pair-wise similarity of entities from different KGs can be derived based on the distance between them in the vector space, which is then used to predict and determine the alignment of entities.

While numerous entity alignment approaches for knowledge graphs have been proposed in recent years, very few studies carry out experiments to systematically compare, evaluate, and analyse different state-of-the-art approaches. More importantly, it has been overlooked by a wide range of approaches that existing benchmark datasets oversimplify

the challenges of entity alignment in the real world. For one thing, entities are densely connected between KGs in current datasets, making it much easier for entity alignment approaches to achieve good performance. For another, in the existing datasets, each entity in the source KG has exactly one counterpart in the target KG, which is a very unrealistic scenario in real life as KGs are created for different purposes in different domains. Based on the above observations, this research project aims to study the state-of-the-art entity alignment approaches by comprehensively reviewing the algorithm and framework of each approach, deriving datasets that are closer to the real-world setting, conducting a series of exploratory experiments to discuss and analyse their strengths and weaknesses, and proposing potential extensions to the current work as well as promising research directions for future work. Given the context and time limit of this project, three approaches, including MTransE [9], GCN-Align [1], and MuGNN[2], are selected and investigated, leaving room for further exploration in the future.

The rest of this report is organised as follows. Chapter 2 reviews and summarises related literature regarding two subfields: knowledge graph embedding models and knowledge graph entity alignment approaches. Chapter 3 defines the studied problem and provides preliminaries of the three approaches investigated in this project. Chapter 4 details the experiments and results and presents critical analysis and discussion. Chapter 5 concludes the project with future research directions.

Chapter 2

Review of Related Literature

In this chapter, we divide the related literature into two subfields: knowledge graph embedding and knowledge graph entity alignment. We start by giving an overview of the existing knowledge graph embedding models (2.1), followed by a review of knowledge graph entity alignment approaches (2.2) in the current literature, including both conventional approaches and embedding-based approaches.

2.1 Knowledge Graph Embedding

Existing KG embedding models can be generally divided into three categories: translational models, semantic matching models, and deep models. In this section, we briefly review the representative work in each category.

Translational Models. TransE [10] is one of the most representative translational models for knowledge graph embedding. It interprets a relationship vector as the translation from the subject entity vector to the object entity vector. TransE is a simple but powerful method that has shown the great capability to model one-to-one relationships and achieve promising results. However, it has been found that TransE has difficulty in effectively modelling more complex relationships, such as one-to-many relationships and many-to-many relationships [11]. To further improve TransE, researchers have proposed a number of models based on TransE, including TransH [12], TransR [13], and TransD [14]. Compared with TransE, these models embed entities and relationships into different embedding spaces by using a distributed representation to separate the relationship vector space from the entity vector space. In this way, the ability of the embedding models to express complex relationships can be enhanced.

Semantic Matching Models. Another important category of KG embedding models is semantic matching models, which leverage similarity-based scoring functions to infer relationship facts. RESCAL [15] and HolE [16] are two examples within this category, which exploit tensor factorisation and represent relationships with matrices. DistMult [17] simplifies RESCAL by forcing the relationship matrices to be diagonal matrices. ComplEx [18] further extends DistMult with complex-valued embeddings in order to model asymmetric relationships in a more effective way. SimplE [19] also restricts relationship embeddings to diagonal matrices, but it extends DistMult by associating each entity with two separate embeddings and each relationship with two separate diagonal matrices so that the model is fully expressive and can model asymmetric relationships well.

Deep Models. Deep models refer to models that use deep learning techniques for KG embedding. These models combine the input data with learnable parameters to discover significant patterns. ConvE [20] is one of the first models that employ convolutional neural networks (CNNs), which uses 2D convolution over embeddings and multiple layers of non-linear features to model knowledge graphs. Recently, graph convolutional networks (GCNs) have also shown promising performance in many studies [21–23]. As an extension of GCNs, R-GCN [24] has been proposed by Schlichtkrull et al., which applies GCNs to deal with highly multi-relational data characteristics of knowledge graphs. Apart from CNNs and GCNs, recurrent neural networks (RNNs) has also been considered and studied by researchers. However, basic RNNs suffer greatly from the limitation that they do not explicitly handle the path alternation of entities and relationships. RSN [25] is developed to tackle such issue, which effectively bridges the gaps between entities based on a skipping mechanism and combines RNNs with residual learning to better capture the relational dependencies within and between KGs.

2.2 Knowledge Graph Entity Alignment

Entity alignment between knowledge graphs has been an active research area for many years. With its prevalence in various applications, there has been an explosion of interest in developing entity alignment approaches to enhance knowledge fusion. This section provides a review of related literature on knowledge graph entity alignment approaches, including conventional approaches and embedding-based approaches.

Conventional Entity Alignment. Conventional approaches deal with entity alignment problem from two main perspectives, namely equivalence reasoning [26, 27] and similarity computation [28–30]. Some recent work improves the alignment accuracy by using statistical machine learning [31, 32] and crowdsourcing [33]. However, it is worth

noting that conventional approaches largely rely on literal information of entities and require massive collaborative efforts. Although they can achieve high alignment accuracy, it cannot be ignored that they suffer from extension inflexibility and high costs of labour and time.

Embedding-based Entity Alignment. As mentioned in the previous section, recent years have witnessed great advancements in the development of embedding models. Such advancements motivate researchers to study and explore embedding-based entity alignment. Compared with conventional approaches, embedding-based approaches require much less human involvement and can be scaled to large KGs more easily as a result.

Many existing approaches employ translational models (e.g., TransE [10]) for entity alignment. Chen et al. propose MTransE, which adopts TransE to train knowledge graphs in separated embedding space and develops an alignment model to construct transitions between vector spaces [9]. IPTransE [34] encodes and unifies entities and relations into a low-dimensional space before mapping the knowledge embeddings into a joint semantic space and implementing an iterative method to improve its alignment performance further. Similarly, BootEA [35] also iteratively adds the newly discovered entity alignments to the training set during the optimisation process. Additionally, it also employs an error correction mechanism to mitigate the impact of error accumulation. Besides, some approaches, such as JAPE [36] and KDCoE [37], incorporate attributes and description information into their algorithms in order to improve the entity embeddings. Exploiting attribute values, Trisedya et al. propose an embedding-based approach that integrates entity structure embedding with attribute character embedding to improve the performance of entity alignment between two KGs [38].

Some recent approaches are developed based on graph neural networks (e.g., GCNs [21]). For instance, Wang et al. present GCN-Align [1], which embeds entities from each knowledge graph into a unified vector space and discovers entity alignment via GCNs. Cao et al. introduce a Multi-channel Graph Neural Network model (MuGNN) [2] to reconcile the structural differences between different KGs and make better use of the seed alignment at the same time. HGCN [39], another GCN-based approach, approximates relationship representations using a small set of aligned entities before incorporating them into entities to learn representations for both entities and relationships iteratively. To better explore and capture complex relationship information in multi-relational knowledge graphs, Wu et al. develop RDGCN [11], which incorporates relationship information into entity representations by allowing multiple rounds of interactions between the primal entity graph and corresponding dual relation graph. It further extends GCNs with highway gates so that neighbourhood structural information can be integrated.

Chapter 3

Preliminaries

A review of related literature in Chapter 2 highlights a number of important work in the field of knowledge graph entity alignment. However, given the context of this research project, we select three of them, namely MTransE, GCN-Align, and MuGNN, to investigate by implementing a series of experiments and analysing their performances. Before presenting a detailed discussion and analysis of the experiments, we first provide some preliminaries in this chapter. We start with the problem definition (3.1), which is followed by the explanation of two mainstream KG embedding models (3.2) and the three KG entity alignment approaches (3.3).

3.1 Problem Definition

Knowledge graphs store knowledge of real-world entities in triples. A KG consists of both relationship triples and attribute triples. Relationship triples represent the relationship between entities in the form of *(subject entity, relationship, object entity)*, and attribute triples represent attributes of entities in the form of *(entity, attribute, value)*. For instance, in Wikidata, *(Albert_Einstein, influenced, Leo_Szilard)* is a relationship triple where *influenced* is the relationship between *Albert Einstein* and *Leo Szilard*, and *(Albert_Einstein, date of birth, 1879-03-14)* is an attribute triple where *date of birth* is an attribute of *Albert Einstein* and *1879-03-14* is the attribute value. Both relationship triples and attribute triples contain essential information about the entities.

Formally, we represent a knowledge graph as $G = (E, R, A, T_r, T_a)$, where E, R, and A are sets of entities, relationships, and attributes, respectively; T_r and T_a denote the sets of relationship triples and attribute triples. Given two KGs $G = (E, R, A, T_r, T_a)$ and $G' = (E', R', A', T'_r, T'_a)$, the task of entity alignment can be defined as discovering

entities in G and G' that represent the same real-world object. In many cases, a set of pre-aligned entities $A_e^s = \{(e, e') \in E \times E' | e \leftrightarrow e'\}$ and a set of pre-aligned relationships $A_r^s = \{(r, r') \in R \times R' | r \leftrightarrow r'\}$ with $\leftrightarrow$ representing equivalence, are known beforehand as seed alignments and are used as training data.

3.2 Knowledge Graph Embedding Models

TransE [10] and GCN [21] are two mainstream models that are frequently used by various entity alignment approaches to learn and generate the embeddings for entities. Here we provide a brief description of these two models.

3.2.1 TransE

TransE is a typical triple-based embedding model that captures the local semantics of relationship triples and aims to preserve the structural information of entities. Specifically, it models relationships as translations operating on the low-dimensional representations of the entities. Given a relationship triple (s, r, o), TransE suggests that the embedding of the object entity o should be close to the embedding of the subject entity s plus the embedding of the relationship r, i.e., $\boldsymbol{s} + \boldsymbol{r} \approx \boldsymbol{o}$. Here bold-face letters denote the corresponding vector representations. In this way, entities with similar neighbourhood structures should have a closer representation in the embedding space. The energy of the relationship triple is

$$\phi(s, r, o) = ||\boldsymbol{s} + \boldsymbol{r} - \boldsymbol{o}|| \tag{3.1}$$

where $|| \cdot ||$ is the L1-Norm or L2-Norm of vectors. TransE minimises the margin-based loss function by a pre-defined margin to separate positive triples from negative triples.

3.2.2 GCN

The graph convolutional network (GCN) is a type of neural networks that is well suited for modelling graph-structured data. It operates on graph data directly and generates node-level outputs by encoding the information about the neighbourhood of nodes. Concretely, the inputs of a GCN include feature vectors of each node in the knowledge graph and an adjacency matrix describing the graph structure, based on which the GCN aims to learn a function of the features on the input graph and produce a new feature matrix as the output. A GCN model normally consists of multiple stacked GCN layers, and the

typical propagation rule from the l^{th} layer to the $(l+1)^{th}$ layer is

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \tag{3.2}$$

where σ is the activation function such as $Tanh(\cdot)$ and $ReLU(\cdot)$; $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ with $\boldsymbol{A}$ being the adjacency matrix that represents the structural information of the graph and $\boldsymbol{I}$ being the identity matrix; $\tilde{\boldsymbol{D}}$ is the diagonal node degree matrix of $\tilde{\boldsymbol{A}}$; $\boldsymbol{W}^{(l)}$ is the learnable weight matrix in the l^{th} layer. $\boldsymbol{H}^{(l+1)}$ denotes a new feature matrix which is the output from the l^{th} layer.

3.3 Knowledge Graph Entity Alignment Approaches

In this section, we present a detailed review of the algorithm and framework of MTransE [9], GCN-Align [1], and MuGNN [2] in order to lay a solid foundation for the experiments and analysis in the next chapter.

3.3.1 MTransE

MTransE is a translation-based approach originally designed for multilingual entity alignment. The idea of MTransE can be illustrated by Figure 3.1. It first adopts TransE [10] to compute the embeddings for each KG. Then, it provides transitions for each embedding vector to its counterparts in other spaces, while preserving the key properties of each knowledge graph.

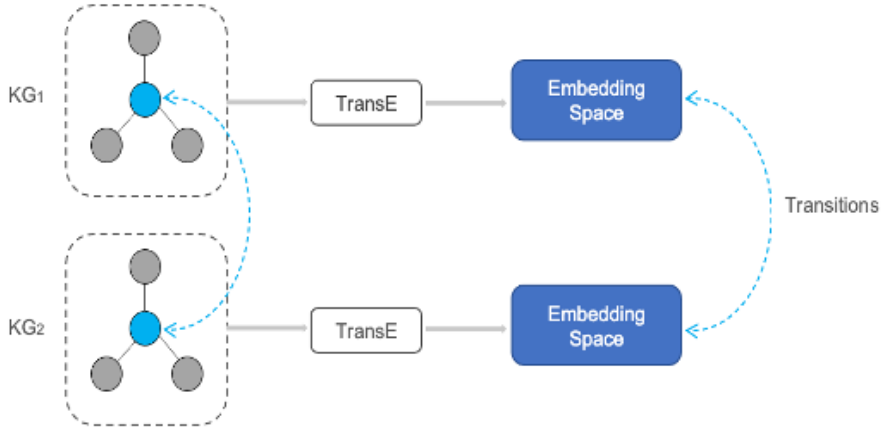

FIGURE 3.1: MTransE

MTransE consists of two components, namely the knowledge model and the alignment model. Specifically, the knowledge model encodes entities and relationships of each knowledge graph in a separated embedding space by using TransE. As for the alignment

model, it aims to learn cross-lingual transitions from a set of pre-aligned relationship triples. The loss function of the alignment model is

$$\mathcal{L}_A = \sum_{(T_r, T_r') \in \delta(L_i, L_j)} S_a(T_r, T_r') \tag{3.3}$$

where (T_r, T_r') denotes an aligned relationship triple pair and $\delta(L_i, L_j)$ denotes the alignment set that contains the pre-aligned relationship triple pairs between language L_i and language L_j. MTransE considers three different techniques to compute the alignment score $S_a(T_r, T_r')$, including distance-based axis calibration, translation vectors, and linear transformations. Out of these three techniques, deducing linear transformations between the embedding spaces achieves the best performance and the corresponding alignment score is

$$S_a = ||\boldsymbol{M}_{ij}^e \boldsymbol{s} - \boldsymbol{s'}|| + ||\boldsymbol{M}_{ij}^e \boldsymbol{o} - \boldsymbol{o'}|| \tag{3.4}$$

with $\boldsymbol{M}_{ij}^e$ as the linear transformation on entity vectors from L_i to L_j.

Combining the knowledge model and the alignment model, MTransE minimises the overall loss function, which is the weighted sum of the two models' loss. It is noteworthy that negative sampling is not employed, since the authors did not find any noticeable impact on the results of their experiments.

3.3.2 GCN-Align

GCN-Align is one of the first GNN-based approaches for knowledge graph alignment problem, and its framework is presented in Figure 3.2. Given two KGs and a set of pre-aligned entity pairs, it trains graph convolutional neural networks (GCNs) to embed entities from the two KGs into a unified vector space and identify aligned entities based on the distance between them.

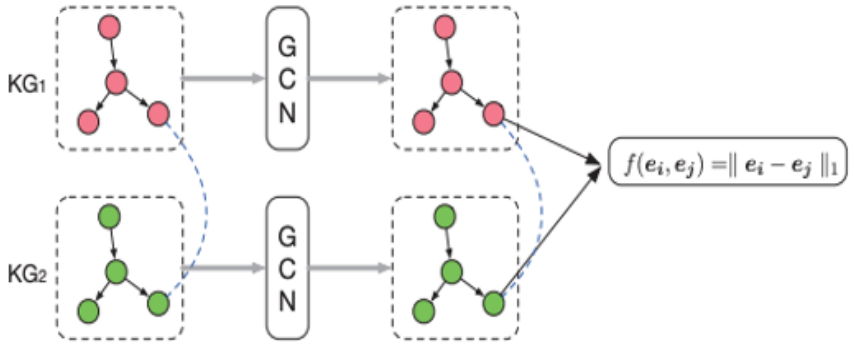

FIGURE 3.2: GCN-Align [1]

One of the key contributions of GCN-Align is that it effectively combines entity relations and entity attribute values to improve the results of KG alignment. To exploit both structure and attribute information of entities, GCN-Align assigns two feature vectors, including a structure vector and an attribute vector, to each entity in the GCN layers. Then, the convolution computation is redefined as:

$$[\boldsymbol{H}_s^{(l+1)}; \boldsymbol{H}_a^{(l+1)}] = ReLU(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}[\boldsymbol{H}_s^{(l)}\boldsymbol{W}_s^{(l)}; \boldsymbol{H}_a^{(l)}\boldsymbol{W}_a^{(l)}]) \tag{3.5}$$

where $\boldsymbol{H}_s^{(l)}$ and $\boldsymbol{H}_a^{(l)}$ are structure and attribute feature matrices of all entities; $\boldsymbol{W}_s^{(l)}$ and $\boldsymbol{W}_a^{(l)}$ are weight matrices for structure features and attribute features respectively; and $[;]$ denotes the concatenation of two matrices. Considering that KGs are directed graphs with entities connected by different types of relations, it designs a special way of computing $\boldsymbol{A}$ instead of directly using the adjacency matrix. Let $a_{ij} \in \boldsymbol{A}$ indicate the extent of alignment information propagating from the i^{th} entity e_i to the j^{th} entity e_j in knowledge graph G. Then,

$$a_{ij} = \sum_{<e_i,r,e_j>\in G} ifun(r) + \sum_{<e_j,r,e_i>\in G} fun(r) \tag{3.6}$$

where $fun(r)$ and $ifun(r)$ are two measures proposed by the authors, namely functionality and inverse functionality. Concretely, $fun(r)$ and $ifun(r)$ are the number of subject entities and object entities of relationship r divided by the number of triples of r, respectively.

As for entity alignments between KGs, they are predicted based on the distances between the entities in the GCN representation space, where the distances are expected to be small for equivalent entity pairs and large for non-equivalent pairs. A set of pre-aligned entities is used as training data to embed equivalent entities as close as possible in the representation space. The model is trained by minimising the margin-based loss functions for structure embedding and attribute embedding separately.

3.3.3 MuGNN

Although GCN-Align achieves decent performance in entity alignment tasks, Cao et al. point out the it fails to consider the structural differences between the KGs [2]. To reconcile the structural heterogeneity of KGs and make better use of the seed alignments, they propose another GNN-based approach, MuGNN. As is shown in Figure 3.3, KG completion and multi-channel graph neural network are two key steps of the MuGNN framework.

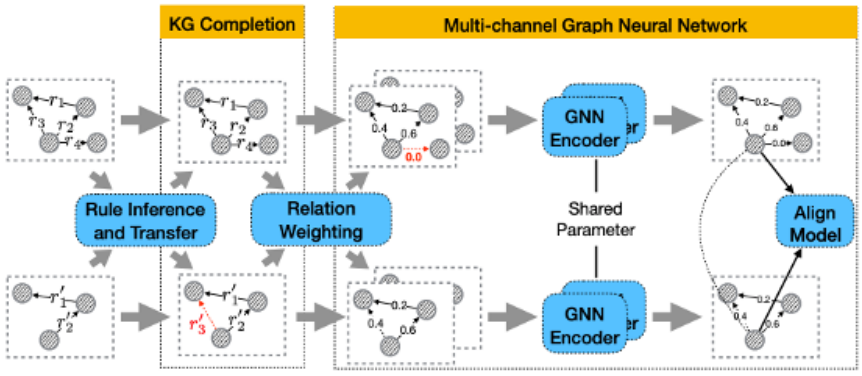

FIGURE 3.3: MuGNN [2]

KG completion aims to reconcile the structural differences between KGs through rule inference and transfer. It employs a popular rule mining system, AMIE+ [40], to induce rules from each KG before transferring them between KGs based on the assumption that knowledge can be generalised into different KGs regardless of domains. The rule sets are then grounded on the corresponding KG for consistent completion. To better illustrate the idea behind KG completion, here is an example. As we can see in Figure 3.4, the red dashed lines and ellipse in KG_2 illustrates the structural differences between the two KGs, making KG_2 more informative than KG_1. After obtaining the rule $(x, province, y) \wedge (y, dialect, z) \Rightarrow (x, dialect, z)$ from KG_2, we can transfer it to KG_1 based on the aligned relationships: *province* and *dialect*. In this case, a new relationship triple $(Jilin\ City, dialect, Northeastern\ Mandarin)$ can be derived.

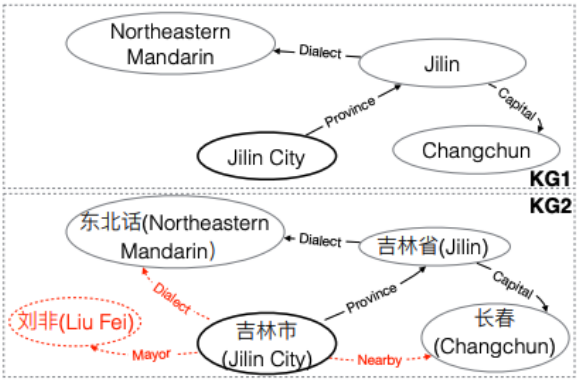

FIGURE 3.4: Illustration of KG Completion in MuGNN (dashed lines and ellipse in red denote the additional information in KG_2) [2]

The main objective of multi-channel graph neural network is to encode each KG via multiple channels. Specifically, it is comprised of three components, including relation weighting, multi-channel GNN encoder, and align model. Relation weighting is designed to generate weighted connectivity matrix which will be the input of GNN encoder. The

model considers two types of structural differences among different KGs: the missing relationships due to the incompleteness nature of KGs and the exclusive entities due to different construction demands of applications. These two types of differences are reconciled by using two channels of GNN encoder for each KG. As a result, two adjacency matrices are generated for each KG: $\boldsymbol{A_1}$ based on KG self-attention and $\boldsymbol{A_2}$ based on cross-KG attention. Following the graph attention networks [41], KG self-attention aims to make better use of the seed alignments with a_{ij} in $\boldsymbol{A_1}$ defined as

$$a_{ij} = softmax(c_{ij}) = \frac{exp(c_{ij})}{\sum_{e_k \in N_{e_i} \cup e_i} exp(c_{ik})} \tag{3.7}$$

where $e_k \in N_{e_i} \cup \{e_i\}$ refers to neighbours of e_i with self-loop and c_{ij} refers to the attention coefficient. Regarding cross-KG completion, it aims to capture the common subgraph of the two KGs and prune exclusive entities with a_{ij} in $\boldsymbol{A_2}$ defined as

$$a_{ij} = \max_{r \in R, r' \in R'} \mathbf{1}((e_i, r, e_j) \in T_r) sim(r, r') \tag{3.8}$$

where $\mathbf{1}(\cdot)$ is 1 if holds true, else 0; $sim(r, r')$ is the inner product similarity between relationship types. Given $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$, the multi-channel GNN encoder is constructed by stacking multiple GCN encoders:

$$MultiGNN(\boldsymbol{H}^l; \boldsymbol{A}_1, \boldsymbol{A}_2) = Pooling(\boldsymbol{H}_1^{l+1}, \boldsymbol{H}_2^{l+1}) \tag{3.9}$$

where $\boldsymbol{H}_i^{l+1}$ denotes the node representations in the $(l+1)^{th}$ layer and i^{th} channel; $Pooling(\cdot)$ refers to the average pooling techniques. Then, the align model embeds the two KGs into a unified vector space using pre-aligned entities and relationships. Since new triples are added during the rule grounding process, triple loss is also introduced to hold the grounded rules as valid in the unified vector space. Hence, the model is optimised using the margin-based loss function that consists of both alignment loss and triple loss.

Chapter 4

Experiments and Analysis

In Chapter 4, we study and compare MTransE, GCN-Align, and MuGNN through four exploratory experiments on two datasets. We begin by providing an overview of the experiment settings and datasets used (4.1) before moving on to the four experiments, focusing on the impact of seed alignments (4.2), efficiency analysis (4.3), the impact of attribute triples (4.4), and the impact of entity alignment on the internal semantics of each KG (4.5), respectively. We end the chapter with a summary of the results and findings from the experiments (4.6).

4.1 Overview

We start with a brief overview of the experiment settings. We use different sizes of seed alignments as training data in Experiment 1 to compare the three approaches' performances and investigate the impact of seed alignments. The maximum memory usage of each approach is reported for efficiency analysis in the second experiment. In Experiment 3, we compare the three approaches with and without attribute triples. Additionally, different ways of using attribute triples are explored and discussed. The last experiment is about the effect of entity alignment on each KG's internal semantic, which is measured by link prediction for translation-based approaches and node classification for GNN-based approaches. For each experiment, we explain how it is designed and executed and present the result with a detailed analysis.

As for the datasets used in the experiments, the entity alignment approaches are evaluated on three publicly available knowledge graphs, namely DBpedia (DBP) [4], YAGO [5], and Wikidata (WD) [6]. Specifically, we run the approaches to align entities from

TABLE 4.1: Dataset Statistics

Dataset		Entities	Attributes	Relationships	Attribute Triples	Relationship Triples
DBP-YAGO	DBP	58,858	85	126	173,520	87,676
	YAGO	60,228	38	53	186,328	66,546
DBP-WD	DBP	84,911	257	288	221,591	203,502
	WD	86,116	501	202	223,232	198,797

DBP with the ones from YAGO and WD, respectively. Then, the aligned entity pairs discovered by the approaches are compared with the ground truth datasets [1], **DBP-YAGO** and **DBP-WD**. DBP-YAGO contains 15,000 aligned entities, 43 aligned relationships, and 29 aligned attributes; DBP-WD contains 50,000 aligned entities, 49 aligned relationships, and 28 aligned attributes. The statistics of the two datasets are included in Table 4.1. It is worth noting that to better mirror the real-world challenges and difficulties regarding entity alignment between KGs, these two datasets are derived from the existing benchmark datasets, DWY100K(DBP-YAGO) and DWY100K(DBP-WD) [35] respectively, by removing 30% of the aligned entities from the knowledge graphs. We do not directly use the existing benchmark datasets because the existing ones oversimplify the entity alignment challenges under real-life scenario. For instance, in DWY100K, entities are very densely connected to each other, and for every entity in the source KG, we can always find exactly one counterpart in the target KG [42]. In this case, an entity alignment approach can achieve decent performance by simply aligning each entity in the source KG with the most similar one in the target KG. Further, in the real world, KGs are created for a wide range of purposes and problem domains. As a result, they contain entities that other KGs do not possess. By using datasets that are closer to real-world settings, we aim to better compare and analyse each approach's strengths and weaknesses.

4.2 Experiment 1: Impact of Seed Alignments

Most of the existing approaches use 30% of the seed alignments as training data and leave the rest 70% for testing. To have a better understanding of the three approaches and their sensitivity to the usage of seed alignments, we compare their performance using different sizes of seed alignments in Experiment 1.

[1] http://downloads.dbpedia.org/2016-10/links/

4.2.1 Method

To investigate the impact of seed alignments on the three approaches, we use different sizes of seed alignments as training data. Specifically, we gradually increase the size of seed alignments from 10% to 50%. As for the evaluation metrics, by convention, the performance of the entity alignment approaches is evaluated by *Hits@N*, which refers to the proportion of correctly aligned entities ranked in the top N predictions. We report *Hits@1* and *Hits@10* on each dataset. Higher *Hits@N* indicates better performance.

4.2.2 Results and Discussion

Figure 4.1 shows the result of Experiment 1. As we can see, all three approaches perform better as the size of seed alignments increases, which is in line with our intuition, and a slower growth trend in their performances can be observed with 40% and 50% of seed alignments. MuGNN consistently outperforms the other two approaches on both datasets. Compared with the other GNN-based approach, GCN-Align, MuGNN explicitly completes knowledge graphs by mining, inferring, and transferring rules between the two KGs to alleviate the negative impact of the heterogeneity of KG structures, which has not been considered by GCN-Align. Additionally, although both GCN-Align and MuGNN employ graph convolutional networks to encode KGs, MuGNN designs and utilises multi-channel GNN encoders for different types of structural differences to make the most of seed alignments. Nonetheless, one advantage of GCN-Align over MuGNN is that it harnesses both relationship triples and attribute triples, which proves to be an effective way to improve its performance. Apart from that, as rule inference and transfer is the key step in the MuGNN framework, proper rule mining is essential. However, it can be easily overlooked that rule mining might not work well on domain-specific datasets where there are minimal number and types of relationships, and the entity alignment performance might be compromised in this case. *LinkedGeoData* [43] and *Geonames* [2] are two examples of such datasets, which contain mainly geographical data with limited number of location-related relationships. Moreover, while GCN-Align only needs a set of pre-aligned entities as the training data, MuGNN requires a set of pre-aligned entities and relationships. This indicates that the rule transfer might fail to work when there is no aligned-relationship available, and the size of the detected rules will be much smaller in KGs where the structure of KGs is sparser. It is also noteworthy that the inferred rules do not hold in all cases. Thus, the confidence and quality of the rule grounding process is of great importance to the overall performance of MuGNN, which has not been carefully considered and discussed by the authors.

[2] http://www.geonames.org/ontology/

On the other hand, the performance of MTransE is significantly worse than the other two approaches, which can be analysed from several aspects. One important reason is that MTransE embeds KGs using TransE, which is constrained by the assumption that the embedding of the object entity should be close to the embedding of the subject entity plus the embedding of the relationship. This is a strong assumption that can make it inefficient to deal with the complex KG structures and insufficient to generate high-quality entity embeddings. Also, it models the structure of KGs in separate embedding spaces, which might result in more information loss during the transitions between the embedding spaces. Besides, as mentioned in the previous chapter, MTransE removes negative sampling during the training process, which is likely to cause overfitting as embeddings are trained only with positive samples. Since it jointly optimises the knowledge model and alignment model, the overall loss function has to be balanced very carefully, increasing the difficulty of generating promising results. Furthermore, compared with GCN-Align and MuGNN, MTransE neither exploits attribute triples nor induces rules from relationship triples for KG completion, which also explains its relatively inferior results.

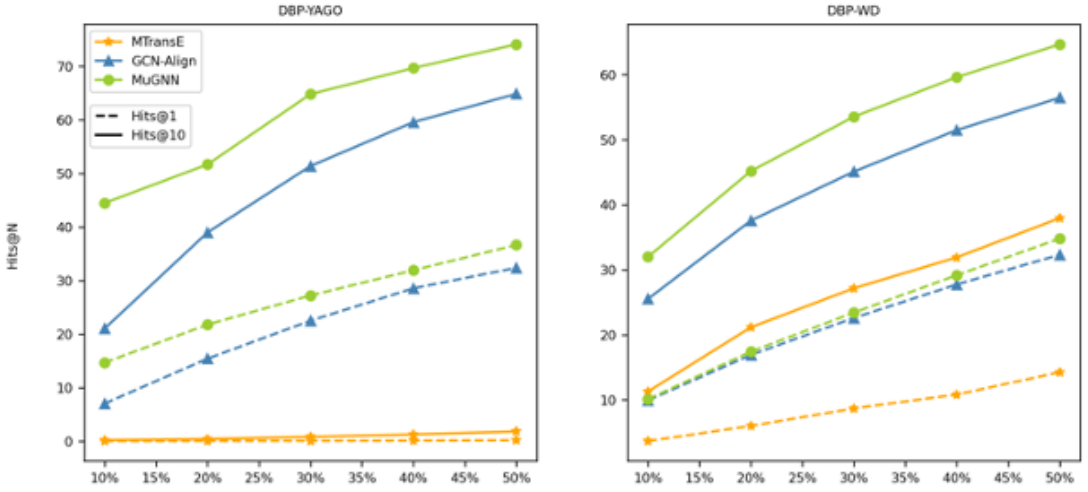

FIGURE 4.1: Experiment 1 Result: Impact of Seed Alignments

4.3 Experiment 2: Efficiency Analysis

To provide a more comprehensive evaluation, Experiment 2 aims to compare and analyse the efficiency of the three approaches by reporting their maximum memory usage on each dataset.

TABLE 4.2: Experiment 2 Result: Maximum Memory Usage

Approach		**MTransE**	**GCN-Align**	**MuGNN**
Dataset	DBP-YAGO	0.41 GB	1.53 GB	17.82 GB
	DBP-WD	0.50 GB	10.63 GB	34.00 GB

4.3.1 Method

All three approaches split 30% of the entity seed alignments as training data leaving the rest for testing in the original papers. In this experiment, we hereby report the maximum memory usage of the approaches on each dataset using 30% entity alignments for training. We admit that different parameter settings, such as the learning rate and the number of epochs, might impact the memory usage of different approaches. Nevertheless, the main objective of this experiment is to present an overview of their efficiency by adopting the parameter settings reported in their original papers.

4.3.2 Results and Discussion

The result of Experiment 2 is listed in Table 4.2. As we can see from the results, due to the more straightforward design of algorithms and more lightweight model complexity, MTransE has the least maximum memory usage among the three approaches, followed by GCN-Align. However, as discussed in the previous section, the performance of MTransE is significantly worse than the other two approaches, especially on the DBP-YAGO dataset. On the other hand, although MuGNN outperforms the other two approaches in Experiment 1, Table 4.2 shows that it requires much more memory, especially on larger datasets like DBP-WD. Although running MuGNN on GPU is an option provided by the authors in their original code, it is worth mentioning that it cannot work in practice due to the memory limitation and all experiments on MuGNN in this project have been run on CPU as a result. Furthermore, along with the large memory usage, the time cost of running experiments on MUGNN is also very high. For instance, MUGNN took 16 hours to generate results on the DBP-WD dataset, whereas GCN-Align only took 1 hour. Therefore, compared with MuGNN and MTransE, GCN-Align can be an efficient and effective candidate to deal with large scale datasets. Last but not least, though only three approaches are investigated in this project, the result of Experiment 2 still implies that having high efficiency on large-scale datasets while being able to achieve promising performance can be a challenge for the state-of-the-art approaches.

4.4 Experiment 3: Impact of Attribute Triples

Experiment 3 is designed as an exploratory experiment to explore and compare the effect of attribute triples on the performance of each approach.

4.4.1 Method

We compare the three approaches with and without attribute triples. Given that only GCN-Align takes attribute triples into consideration in its framework whereas the other two approaches are based on relationship triples only, here is how we design the experiment. For MTransE and MuGNN, which originally only use relationship triples, we take a simple exploitation method to treat the attribute triples as relationship triples. For GCN-Align, which originally exploits attribute triples, we report the result of the structure-only variant of GCN-Align, which only uses relationship triples to perform structure embedding. Moreover, inspired by JAPE [36], for GCN-Align, we carry out an additional experiment as an extension to compare two ways of using attribute triples: one is using the original value of attributes, and the other is using attribute types by categorising the attribute values into 4 data types, including String, Integer, Double, and Date. Similar to Experiment 1, we report *Hits@1* and *Hits@10* on each dataset as the evaluation metrics.

4.4.2 Results and Discussion

The results of Experiment 3 are shown in Figure 4.2 and Figure 4.3 with details included in Table 4.3. Figure 4.2 indicates that considering attribute information can be a useful way to improve the results of entity alignment in most cases. We start with a discussion about the two relationship-based approaches, MTransE and MuGNN. As displayed in Figure 4.2, a significant improvement in the performance of both approaches can be observed on the DBP-YAGO dataset, where there is limited number of relationship triples. This suggests that for approaches that only use relationship triples, a simple exploitation method like treating attribute triples as relationship triples can be considered as a way to enhance the overall performance when only a small number of relationship triples are available. However, when we look at the result on the DBP-WD dataset, the two approaches perform differently. While there is a relatively smaller improvement in MuGNN's performance, the performance of MTransE is compromised with attribute triples treated as relationship triples. For one thing, as the KG structures become more complex and possess more relationship and attribute triples, TransE-based approaches are prone to learn very similar embeddings for different entities [12, 13], resulting in the

TABLE 4.3: Experiment 3 Result: Impact of Attribute Triples

Approach	DBP-YAGO		DBP-WD	
	Hits@1	**Hits@10**	**Hits@1**	**Hits@10**
MTransE (w/o Attr)	0.07	0.81	8.69	27.15
MTransE (w/ Attr)	1.20	3.15	7.86	20.57
GCN-Align (w/o Attr)	18.74	47.11	21.29	42.33
GCN-Align (w/ Attr Types)	19.66	47.93	21.57	42.81
GCN-Align (w/ Attr Values)	22.50	51.37	22.59	45.07
MuGNN (w/o Attr)	27.20	64.80	23.41	53.51
MuGNN (w/ Attr)	60.46	78.19	33.05	55.41

deficiency in generating embeddings of high quality. For another, as discussed previously, MTransE does not employ the negative sampling technique that has been widely used in a number of translation-based approaches and proved to be very valuable for structure embeddings [35, 36, 44]. Only using positive samples in the training process makes it prone to overfitting. This is more likely to happen when the method to exploit of attribute triples is designed to be very simple. On the other hand, although simply treating attribute triples as relationship triples can further improve MuGNN's performance on both datasets, it is worth noting that due to high computational costs and model complexity, it might take even longer to complete the experiments. For instance, it took six days to generate the results on the DBP-WD dataset, which, again, reflects its efficiency issue.

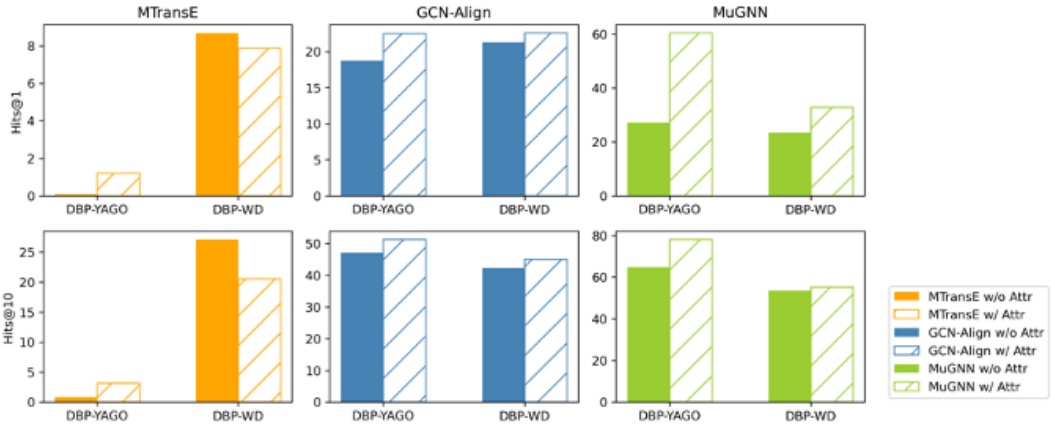

FIGURE 4.2: Experiment 3 Result: Impact of Attribute Triples

Regarding GCN-Align, which originally considers attribute triples in its algorithm, we compare three variants of the approach: (1) GCN-Align without attribute triples, (2) GCN-Align using attribute values, and (3) GCN-Align using attribute types. The result can be found in Figure 4.3. Overall, it validates the usefulness of the attribute information as using either attribute values or types can achieve a better performance than only performing structure embedding. When we have a closer look at the result and compare the two ways of using attribute triples, GCN-Align using attribute types

has a slightly better performance than GCN-Align with structure embedding only. By contrast, GCN-Align using attribute values outperforms the other two variants, which is also in accord with our intuition as it provides more attributional information than using attribute types. However, it has to be admitted that exploiting attribute triples by using attribute values requires a much larger vocabulary.

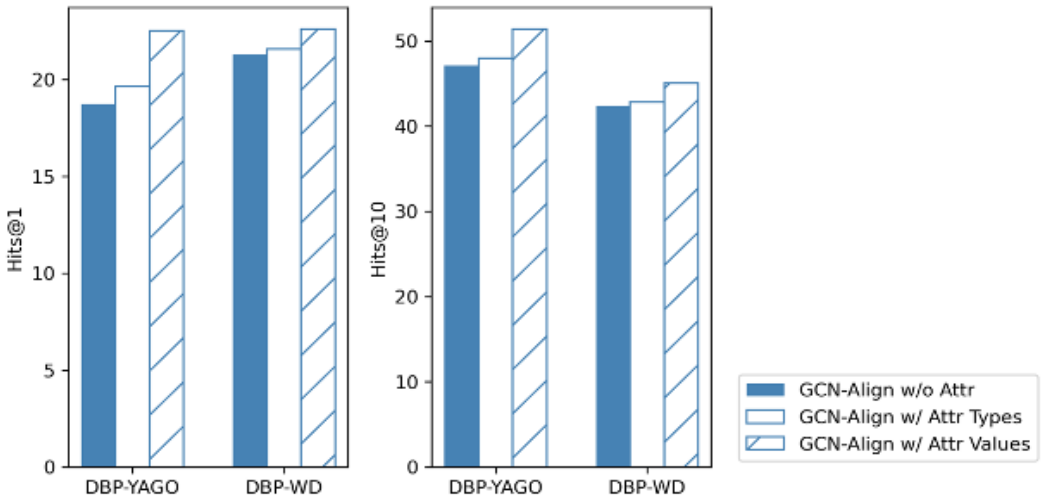

FIGURE 4.3: Experiment 3 Result: Comparison between Using Attribute Values and Attribute Types on GCN-Align

4.5 Experiment 4: Impact of Entity Alignment on the Internal Semantics of Each Knowledge Graph

To further evaluate and compare the three approaches, Experiment 4 aims to investigate and analyse how each approach's alignment model affects the learning results of KG embedding, i.e., the effect of entity alignment on KGs internal semantics.

4.5.1 Method

The effect of entity alignment on the internal semantics of each KG can be measured by downstream applications. In this experiment, we follow two standard tasks: link prediction for translation-based approaches [10] and node classification for GNN-based approaches [21]. Link prediction is to predict the missing object entity given the subject entity and the relationship. The evaluation protocol for link prediction can be explained in two steps. In the first step, we corrupt each relationship triple by replacing its object entity with all the other possible entities in the dataset. In the second step, the corrupted triples are ranked in ascending order based on the plausibility score ($\boldsymbol{s} + \boldsymbol{r} - \boldsymbol{o}$). In this case, valid triples are expected to have smaller plausibility scores than

TABLE 4.4: Experiment 4 Result: Impact of Entity Alignment on KGs Internal Semantic

Approach		**MTransE**	**GCN-Align**	**MuGNN**
		Link Prediction (Hits@10)	**Node Classification (Precision)**	**Node Classification (Precision)**
Dataset	DBP-YAGO	98.03	78.12	43.75
	DBP-WD	84.23	66.87	51.14

invalid ones. Following the convention, we use *Hits@10* as the evaluation metric for the link prediction task. As for node classification, it aims to classify the nodes and determine their labels based on the labelled nodes in their neighbourhood. Under the current experiment setting, it means given the embedding of an entity, we train a simple classifier to predict the corresponding entity type. Specifically, we employ a support vector classifier and evaluate it using two-fold cross-validation. We report the percentage of correctly classified entities for the node classification task.

4.5.2 Results and Discussion

The result of Experiment 4 is included in Table 4.4. Ideally, entity alignment approaches are expected to achieve good alignment performance while being able to preserve the internal representation of KGs. Nonetheless, it is understandable that better performances in entity alignment might compromise the internal semantics of each KG. This is also indicated by the result of this experiment. Specifically, compared with MTransE and GCN-Align, MuGNN is more alignment-oriented, which is clearly reflected by the idea and framework behind the approach. As discussed previously, one of the critical objectives of MuGNN is to reconcile the structural differences between KGs. It explicitly completes KGs through rule inference and transfer and prunes exclusive entities through specially designed relation weighting schemes. As a result, new triples are added, and the structure of each KG is changed, which explains its relatively inferior result in this experiment. On the other hand, MTransE strikingly outperforms the other two approaches in Experiment 4, which is in accord with one of the original intentions stated by the authors. That is, to preserve the essential properties and characterisation of each KG well. However, given its performance in Experiment 1, it also implies that, in MTransE, the internal representation of each KG is preserved "too well" to address the entity alignment problem between KGs effectively. This might be caused by an imbalance between the loss of the knowledge model and the alignment model during the joint optimisation process. In general, the result of Experiment 4 reveals that designing a proper alignment model that can achieve good performance in entity alignment and preserve the internal semantics of each KG at the same time remains a challenge for future work.

TABLE 4.5: Comparison of Three Approaches

	MTransE	**GCN-Align**	**MuGNN**
Embedding Model	Translation-based	GNN-based	GNN-based
Alignment Model	Transition	Margin-based	Margin-based
Usage of Attribute Triples	✗	✓	✗
Distance Measure	Euclidean Distance	Manhattan Distance	Cosine Similarity
Pros	1) Simple and scalable 2) Key properties of each KG preserved	1) Scalable and effective 2) Attribute triples exploited to improve accuracy	1) Outperformance over the other two approaches 2) Structure heterogeneity of KGs carefully considered 3)Further improvement with simple exploitation of attribute triples
Cons	1) Significant degradation in performance under real-life challenges 2) Prone to overfitting 3) Limited by the TransE-based embedding model	1) Structure heterogeneity of KGs not carefully considered 2) A very large vocabulary required	1) Less scalable and efficient 2) Degradation in performance when there are no/few rules 3) Pre-aligned relationships required

4.6 Comparison and Summary

In this section, we summarise the key findings in the experiments and provide an overall comparison of the three entity alignment approaches (see Table 4.5).

Previous literature shows that MTransE is a simple and scalable approach that works well on most existing benchmark datasets. However, we observe a significant degradation in its performance in our experiments. The limitations of TransE-based embedding models and the absence of the negative sampling technique are two important reasons. There is no doubt that TransE is one of the most frequently used KG embedding models in many existing entity alignment approaches. Yet, as discussed in Chapter 2, there are many unexplored KG embedding models for the entity alignment problem, such as TransH [12], HolE [16], and ConvE [20]. This highlights a potential research direction that is worth further exploration.

Compared with MTransE, GNN-based approaches achieve better and more robust performance. Specifically, MuGNN outperforms the other two approaches by employing rule inference and transfer and using multi-channel GNN encoder to reconcile structural differences between KGs for better alignment performance. Although it originally does not consider attribute triples, a simple exploitation method by treating attribute triples as relationship triples further improves its performance. This not only validates the usefulness of attribute triples, but also points out a direction for future work to propose approaches that effectively leverage attribute triples. Nevertheless, the limitations of MuGNN cannot be ignored. For one thing, pre-aligned relationships between the KGs required by MuGNN might not always be available. Additionally, the performance of the approach greatly relies on the quality of the rule mining, inference, and transfer

process, which requires careful consideration and further investigation. For another, due to its high computational costs and model complexity, efficiency analysis reveals its scalability issue on datasets of a large scale. In this case, GCN-Align can be an efficient and effective candidate for entity alignment on larger datasets. With that said, how to develop algorithms that can efficiently achieve competitive alignment performance remains a research question to be answered by future work. Furthermore, the result of the experiment on how the alignment model of each approach affects the internal semantics of KGs provides motivations for future studies to design and propose proper alignment method that can preserve the internal representation of each KG while having promising alignment performance.

Chapter 5

Conclusion and Future Directions

In Chapter 5, we draw conclusions (5.1) and propose future directions (5.2).

5.1 Conclusion

Entity alignment aims to identify equivalent entities from different knowledge graphs, which is a prevalent approach to integrating knowledge from different KGs for better knowledge fusion and inferences. While entity alignment for knowledge graphs has seen vigorous development and become an active research area in recent years, there have been very few experimental studies to explore, compare, analyse, and discuss the state-of-the-art approaches. With such observations, this project provides an empirical evaluation of three state-of-the-art entity alignment approaches by thoroughly reviewing the algorithm and framework of each approach, carrying out exploratory experiments to compare their strengths and weaknesses, and presenting evidence-based analysis and discussions in detail.

5.2 Future Directions

Based on the experimental results and findings, we propose several future research directions in this section, including both the extensions to the current experimental study and suggestions for future work on entity alignment algorithms.

5.2.1 Extensions to the Current Experimental Study

Variety of State-of-the-Art Algorithms. Given the current research project's context and time limit, only three entity alignment algorithms are investigated and discussed. However, a review of related literature in Chapter 2 shows considerable efforts in this research field. Thus, it is of significance to include a wider variety of state-of-the-art algorithms and categorise them based on their core techniques and characteristics so that the comparison and analysis would be more comprehensive and meaningful.

Similarity on the Top K Alignment. It would be interesting to compute the vector similarity and see the similarity on the top K alignments to analyse the similarity measure performed by different methods. Additionally, the confidence of the alignments can be compared via *similarity@1* for further analysis.

Attribute Semantics. For methods that exploit attribute triples, it is also worth investigating and comparing how each method leverages attribute triples. Specifically, we could compute the semantic similarity of the attribute embedding and plot the attributes in a 2D plane to show whether the same attributes, such as person name and date of birth, are clustered together or not.

5.2.2 Suggestions for Future Work on Entity Alignment

Unsupervised Entity Alignment. As discussed in Chapter 4, seed alignments are required as supervision by most existing entity alignment approaches. However, it can be difficult and costly to meet this requirement in the real world. Moreover, it is vulnerable to the quality of the selected entity pairs. Therefore, one important and meaningful research direction is to study and develop unsupervised entity alignment approaches. Leveraging supplementary resources to distil distant supervision such as pre-trained word embedding can be a possible solution [45]. In addition, recent research by Conneau et al. on unsupervised cross-lingual word alignment highlights that orthogonal Procrustes [46] and adversarial training [47] are also worth exploring [48]. Besides, active learning [49] and abductive learning [50] can be two potential ways to reduce the cost and burden of data labelling.

Scalability of Entity Alignment. The experiments and efficiency analysis in the previous chapter reveal that training and testing the existing methods on larger datasets require much more memory and time, which climb polynomially along with the increasing number of entities. In this case, it is very challenging and costly for embedding-based methods to run on very large knowledge graphs. Hence, another opportunity for future work is to explore ways to improve the scalability of entity alignment approaches. Some

hashing techniques, such as Locality-Sensitive Hashing [51] and Information Network Hashing based on Matrix Factorization (INH-MF) [52], might be useful to alleviate the scalability issue.

Multi-modal Entity Alignment. Information associated with each entity can be in various modalities, including texts, images, or even videos, which potentially add another dimension to the development of entity alignment approaches [53, 54]. Due to the fact that only very few studies have attempted to describe and integrate the multi-modal data for entity alignment problems, how to effectively and efficiently exploit multi-modal knowledge is another research question that remains to be answered in the future.

Entity Alignment in the Real World. As discussed previously, current entity alignment approaches work well under the assumption that every entity in one knowledge graph has a counterpart in another knowledge graph. However, this is not the real-world setting since knowledge graphs are created for different purposes and unmatchable entities always exist. Furthermore, only a few entities are densely connected to others in the real-life knowledge graphs, whereas the rest have a very sparse neighbourhood structure. This has also been oversimplified or even neglected by current literature. Therefore, developing entity alignment algorithms suitable for real-world setting is another challenge for future work. Using more advanced graph neural networks, leveraging additional features such as taxonomies, and extracting information from the open Web to enrich entities are potential directions to investigate [21, 24, 55].

Bibliography

[1] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 349–357, 2018.

[2] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. Multi-channel graph neural network for entity alignment. In *ACL*, 2019.

[3] 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.

[4] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.

[5] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

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

[7] Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. 2015.

[8] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

[9] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*, 2016.

[10] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[11] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*, 2019.

[12] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Aaai*, volume 14, pages 1112–1119. Citeseer, 2014.

[13] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015. ISBN 0262511290.

[14] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687–696, 2015.

[15] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Icml*, volume 11, pages 809–816, 2011.

[16] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. *arXiv preprint arXiv:1510.04935*, 2015.

[17] 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.

[18] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. International Conference on Machine Learning (ICML), 2016.

[19] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, pages 4284–4295, 2018.

[20] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. *arXiv preprint arXiv:1707.01476*, 2017.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[22] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

[23] Diego Marcheggiani and Ivan Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*, 2017.

[24] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[25] Lingbing Guo, Zequn Sun, and Wei Hu. Learning to exploit long-term relational dependencies in knowledge graphs. *arXiv preprint arXiv:1905.04914*, 2019.

[26] Hugh Glaser, Afraz Jaffri, and Ian Millard. Managing co-reference on the semantic web. 2009.

[27] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.

[28] Fabian M Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *arXiv preprint arXiv:1111.7164*, 2011.

[29] Simon Lacoste-Julien, Konstantina Palla, Alex Davies, Gjergji Kasneci, Thore Graepel, and Zoubin Ghahramani. Sigma: Simple greedy matching for aligning large knowledge bases. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 572–580, 2013.

[30] Chao Shao, Lin-Mei Hu, Juan-Zi Li, Zhi-Chun Wang, Tonglee Chung, and Jun-Bo Xia. Rimom-im: A novel iterative framework for instance matching. *Journal of computer science and technology*, 31(1):185–197, 2016.

[31] Ahmed El-Roby and Ashraf Aboulnaga. Alex: Automatic link exploration in linked data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1839–1853, 2015.

[32] Robert Isele and Christian Bizer. Learning expressive linkage rules using genetic programming. *arXiv preprint arXiv:1208.0291*, 2012.

[33] Yan Zhuang, Guoliang Li, Zhuojian Zhong, and Jianhua Feng. Hike: A hybrid human-machine method for entity alignment in large-scale knowledge bases. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1917–1926, 2017.

[34] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via joint knowledge embeddings. In *IJCAI*, volume 17, pages 4258–4264, 2017.

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

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

[37] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. *arXiv preprint arXiv:1806.06478*, 2018.

[38] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 297–304, 2019.

[39] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. Jointly learning entity and relation representations for entity alignment. *arXiv preprint arXiv:1909.09317*, 2019.

[40] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. *The VLDB Journal*, 2015. URL https://hal-imt.archives-ouvertes.fr/hal-01699866.

[41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[42] Xiang Zhao, Weixin Zeng, Jiuyang Tang, Wei Wang, and Fabian Suchanek. An experimental study of state-of-the-art entity alignment approaches. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[43] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354, 2012.

[44] Patrick Klein, Simone Paolo Ponzetto, and Goran Glavaš. Improving neural knowledge base completion with cross-lingual projections. Association for Computational Linguistics, 2017.

[45] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. Aligning cross-lingual entities with multi-aspect information. *arXiv preprint arXiv:1910.06575*, 2019.

[46] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[48] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[49] Kun Qian, Lucian Popa, and Prithviraj Sen. Active learning for large-scale entity resolution. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1379–1388, 2017.

[50] Zhi-Hua Zhou. Abductive learning: Towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62(7):76101, 2019.

[51] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *Vldb*, volume 99, pages 518–529, 1999.

[52] Defu Lian, Kai Zheng, Vincent W Zheng, Yong Ge, Longbing Cao, Ivor W Tsang, and Xing Xie. High-order proximity preserving information network hashing. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1744–1753, 2018.

[53] Ye Liu, Hui Li, Alberto Garcia-Duran, Mathias Niepert, Daniel Onoro-Rubio, and David S Rosenblum. Mmkg: multi-modal knowledge graphs. In *European Semantic Web Conference*, pages 459–474. Springer, 2019.

[54] Liyi Chen, Zhi Li, Yijun Wang, Tong Xu, Zhefeng Wang, and Enhong Chen. Mmea: Entity alignment for multi-modal knowledge graph. In *International Conference on Knowledge Science, Engineering and Management*, pages 134–147. Springer, 2020.

[55] Baoxu Shi and Tim Weninger. Open-world knowledge graph completion. *arXiv preprint arXiv:1711.03438*, 2017.