# Cross-Lingual Entity Matching for Knowledge Graphs

by

Hsiu-Wei Yang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

## Author’s Declaration

# Abstract

Multilingual knowledge graphs (KGs), such as YAGO and DBpedia, represent entities in different languages. The task of cross-lingual entity matching is to align entities in a source language with their counterparts in target languages. In this thesis, we investigate embedding-based approaches to encode entities from multilingual KGs into the same vector space, where equivalent entities are close to each other. Specifically, we apply graph convolutional networks (GCNs) to combine multi-aspect information of entities, including topological connections, relations, and attributes of entities, to learn entity embeddings. To exploit the literal descriptions of entities expressed in different languages, we propose two uses of a pre-trained multilingual BERT model to bridge cross-lingual gaps. We further propose two strategies to integrate GCN-based and BERT-based modules to boost performance. Extensive experiments on two benchmark datasets demonstrate that our method significantly outperforms existing systems. We additionally introduce a new dataset comprised of 15 low-resource languages and featured with unlinkable cases to draw closer to the real-world challenges.

# Acknowledgements

I would like to thank my advisor, Professor Jimmy Lin, for giving me the opportunity to work on exciting problems and to collaborate with inspiring researchers. Without his continuing guidance and support, this thesis would not be possible.

I am also grateful to the readers of my thesis, Professor Ihab F. Ilyas and Ming Li, for reviewing my work.

My special thanks go to Peng Shi for all the discussions and his insights that contributed greatly to this thesis.

Finally, I would like to thank all the old and new friends who have made my time at the University of Waterloo an enjoyable experience.

## Dedication

This is dedicated to my wife, Shu-Han, for her unconditional love and support.

# Table of Contents

# List of Figures

# List of Tables

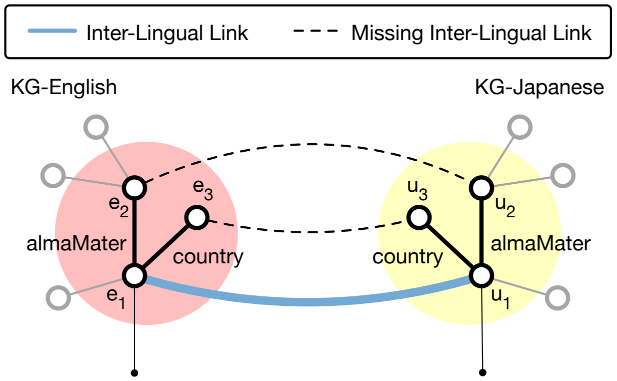
# Chapter 1

# Introduction

A knowledge graph (KG) is a technology that stores and represents real-world knowledge with a graph structure. Specifically, such a structure often employs Resource Description Framework (RDF) as the data model, in which a *fact* is expressed in the form of (*subject, predicate, object*) or (*head, relation, tail*), known as a triple. The constructed topology can facilitate logical reasoning and benefit downstream natural language processing (NLP) tasks, such as question answering [54, 82] and dialogue systems [38, 57]. Motivated by their wide range of applications, a number of large-scale projects of KG construction have been proposed, e.g., Freebase [9], YAGO [91], and DBpedia [8], whose data mostly are extracted from Wikipedia and WordNet [30]. Initially, these KGs only focused on English content. However, the aforementioned knowledge sources actually cover plenty of languages, e.g., Wikipedia has been created in 312 languages;[1] hence, these projects naturally evolved into multilingual KGs [50, 80].

Multilingual KGs typically represent knowledge as separately-structured monolingual KGs. Such KGs are connected by inter-lingual links (ILLs) that align entities with their counterparts in different languages, exemplified by Figure 1.1 (top). These ILLs can be used to enable multilingual data integration and further enhance the existing knowledge in KGs. For instance, the contents documented in resource-poor languages, e.g., Inuktitut, can refer to more information from their counterparts in resource-rich languages, e.g., English. Moreover, when the knowledge originates from a certain culture, in general, the content written in the native language is more comprehensive and accurate. Combined with the explicit semantics of the RDF data model, multilingual KGs are especially valuable for certain cross-lingual NLP tasks, such as machine translation [59] and cross-lingual named entity recognition [23].

[1] https://en.wikipedia.org/wiki/List_of_Wikipedias



| English: University of Toronto | | Japanese: トロント大学 | |
|---|---|---|---|
| Attribute | Value | Attribute | Value |
| Name | University of Toronto | 大学名 | トロント大学 |
| Type | Public University | 学校種別 | 州立 |
| Found Date | 1827-03-15 | 創立年 | 1827 |
| Campus | Ontario | キャンパス | セントジョージ（トロント） |
| Former Name | King's College | 旧名 | キングスカレッジ |
| | ⋮ | | ⋮ |

| Descriptions | |
|---|---|
| The University of Toronto is a public research university in Toronto, Ontario, Canada ··· | トロント大学は、オンタリオ州、トロントに本部を置くカナダの州立大学である ··· |

Figure 1.1: An example fragment of two KGs (in English and Japanese) connected by an inter-lingual link (ILL). In addition to the graph structures (top) consisting of entity nodes and typed relation edges, KGs also provide attributes and literal descriptions of entities (bottom).

Nowadays, the work of knowledge graph completion, i.e., adding new knowledge and detecting the error, heavily relies on automated systems, also known as bots. For example, in the early years of Wikidata project, more than 90% of edits are contributed by bots, and it still exceeds 50% in recent years [75]. However, the quality management of the data massively generated by bots has become a critical issue. In terms of multilinguality, 86% of such bots concentrate on only one language [90], but they are not required to provide multilingual labels or descriptions for non-native speakers' reference. This could result in abundant duplicates in other languages because of the difficulty in cross-lingual verification and hamper multilingual content integration, particularly when the existing facts are documented in lesser-spoken languages. Not to mention that the efficiency of manual verification hardly catches up with the speed of such bots. Therefore, the aim of this thesis is to propose advanced methods of automatic ILL completion.

Precisely, the target task is called cross-lingual entity matching (also known as entity alignment or entity resolution), which is to discover entities from different monolingual KGs that actually refer to the same real-world entities.

Traditional methods for this task apply machine translation techniques to translate entity labels [88]. The quality of mappings in the cross-lingual scenario largely depends on the quality of the adopted translation systems. In addition to entity labels, existing KGs also provide multi-aspect information of entities, including topological connections, relation types, attributes, and literal descriptions expressed in different languages [8, 107], as shown in Figure 1.1 (bottom). The key challenge of addressing such a task thus is how to better model and use provided multi-aspect information of entities to bridge cross-lingual gaps and find more equivalent entities (i.e., ILLs).

Recently, embedding-based solutions [20, 92, 113, 105, 17] have been proposed to unify multilingual KGs into the same low-dimensional vector space where equivalent entities are close to each other. Such methods only make use of one or two aspects of the aforementioned information. For example, Zhu et al. [113] relied only on topological features while Sun et al. [92] and Wang et al. [105] exploited both topological and attribute features. Chen et al. [17] proposed a co-training algorithm to combine topological features and literal descriptions of entities. However, combining the multi-aspect information of entities (i.e., topological connections, relations, and attributes, as well as literal descriptions) remains under-explored.

In this thesis, we extend our work [109], presenting a novel machine learning approach to learn cross-lingual entity embeddings by using all aforementioned aspects of information in KGs, and further introduce a new benchmark dataset to draw closer to the real-world problem. To be specific, we propose two variants of GCN-based models, namely MAN and HMAN, which incorporate multi-aspect features, including topological features, relation types, and attributes into cross-lingual entity embeddings. To capture the semantic relatedness of literal descriptions, we fine-tune the pre-trained multilingual BERT model [25] to bridge cross-lingual gaps. Furthermore, we design two strategies to combine GCN-based and BERT-based modules to make alignment decisions. The experiments on two benchmark datasets show that our method achieves new state-of-the-art results. In addition, we create a new dataset to reflect the practical challenges facing the limited amount of information in low-resource languages and unlinkable cases.

## 1.1 Problem Definition

In a multilingual knowledge graph $\mathcal{G}$, we use $\mathcal{L}$ to denote the set of languages that $\mathcal{G}$ contains and $\mathcal{G}_i = \{E_i, R_i, A_i, V_i, D_i\}$ to represent the language-specific knowledge graph in language

$L_i \in \mathcal{L}$. $E_i$, $R_i$, $A_i$, $V_i$ and $D_i$ are sets of entities, relations, attributes, values of attributes, and literal descriptions, each of which portrays one aspect of an entity. The graph $\mathcal{G}_i$ consists of relation triples $\langle h_i, r_i, t_i \rangle$ and attribute triples $\langle h_i, a_i, v_i \rangle$ where $h_i, t_i \in E_i$, $r_i \in R_i$, $a_i \in A_i$ and $v_i \in V_i$. Each entity is accompanied by a literal description, e.g., $\langle h_i, d_{h_i} \rangle$ and $\langle t_i, d_{t_i} \rangle$, where $d_{h_i}, d_{t_i} \in D_i$.

Given two knowledge graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ expressed in source language $L_1$ and target language $L_2$, respectively, there exists a set of pre-aligned ILLs $I\left(\mathcal{G}_1, \mathcal{G}_2\right) = \left\{(e, u) \,|e \in E_1, u \in E_2\right\}$ which can be considered training data. The task of cross-lingual entity matching is to discover the missing ILLs that connect entities in $\mathcal{G}_1$ with their cross-lingual counterparts in $\mathcal{G}_2$.

## 1.2 Contributions

The main contributions of this thesis are summarized below:

- We propose a novel model with the following characteristics:
  - To the best of our knowledge, this is the first method that employs embedding learning on all the four commonly-used aspects of information, namely topological, relation, attribute, and description, to address this task;
  - We combine feedforward neural networks with GCNs to better capture the information from the relation and attribute features;
  - To match the textual information, we devise a variant of BERT which exploits pairwise training to significantly reduce the computational complexity from polynomial to linear during inference.
- Our model achieves new state-of-the-art performance on two existing benchmark datasets.
- We create a new dataset that is closer to the real-world scenario of cross-lingual entity matching. The benchmark compares the effectiveness of different methods across 15 low-resource languages and tests the capability in NIL prediction.

## 1.3 Thesis Organization

The rest of this thesis is structured as follows: Chapter 2 reviews the fundamental background and related work for understanding our task and the progress of previous solutions. Chapter 3 introduces the details of proposed embedding models and the definition of the learning objective. Meanwhile, we describe how to combine the knowledge learned from graph-based and textual information with the integration strategies. In Chapter 4, we first present the experimental datasets and the evaluation metric. Then, we show the performance of the proposed methods, compared with our baselines, and analyze the effectiveness of each individual aspect through feature ablation. The best methods are further tested on the new dataset to investigate the influence of resource limitation in different languages. Chapter 5 concludes this thesis and discusses future work.

# Chapter 2

# Background and Related Work

## 2.1 Knowledge Graphs

To understand the development of KGs, we first describe the emergence of Semantic Web,[1] which is considered as the next step of the World Wide Web (WWW). Sir Tim Berners-Lee [6] proposed this idea and envisioned the Semantic Web as a web of data, where the goal is to make the meaning in Internet data interpretable for machines. Specifically, it is a set of particular standards that allow the data to be easily processed by machines and shared across all the members of the network. Under the collaboration and efforts of the World Wide Web Consortium[2] (W3C) as well as numerous participants from academia and industry, a series of standards were gradually established. The term *Linked Data* [7] was also coined to refer to the datasets interlinked and built by a set of the most recommended standards.

Resource Description Framework[3] is one of the standards proposed in the early stage and became the basis of the Semantic Web. By applying this framework, knowledge representation is organized as a directed and labeled graph. The elementary building block of an RDF graph is a *triple*, whose typical format is (*subject, predicate, object*). Alternatively, it is also common to express a triple as (*head, relation, tail*) in KG literature. Each element in the triple is either a Uniform Resource Identifier (URI), a literal value, or a blank node. URIs are the technology borrowed from the WWW, i.e., Uniform Resource Locator (URL), to enable global identification in the Semantic Web. Although they share the same syntax,[4] which leads to the same string format,

[1] https://www.w3.org/2001/sw/

[2] https://www.w3.org/

[3] https://www.w3.org/RDF/

[4] https://www.w3.org/Addressing/URL/uri-spec.html

note that a URI is not limited to the purpose of locating a file on the WWW. An example of URIs in DBpedia is *http://dbpedia.org/resource/Bob_Dylan*, where *http* is the *scheme*, *dbpedia.org* indicates the *authority*, and *resource/Bob_Dylan* is the *path* to the target resource. In practice, to further simplify the URIs, a *namespace* may be declared in advance. With the same example, the above URI can be abbreviated as *dbr:Bob_Dylan* after declaring the mapping of source names: "*http://dbpedia.org/resource/*" = "*dbr*".

To exemplify the use of an RDF triple, here we represent a *fact* of real-world knowledge — Bob Dylan's hometown is Hibbing, Minnesota — with (*dbr:Bob_Dylan, dbo:hometown, dbr:Hibbing,_Minnesota*), where *dbr:Bob_Dylan* and *dbr:Hibbing,_Minnesota* are the head and tail *entities*, which are the nodes in the graph, and *dbo:hometown* is the predicate, which is the edge connecting the two nodes and specifying their *relation*. In addition, all these elements can have *types*, and the triple format is still applicable for expressing such information, e.g., (*dbr:Bob_Dylan, rdf:type, dbo:Person*) or (*dbo:hometown, rdf:type, rdf:Property*). However, an element of a triple is also allowed to be a literal value. For example, (*dbr:Bob_Dylan, dbo:birthYear,* **literal**(*1941-01-01*)) represents the fact that Bob Dylan was born in 1941, where the object is a literal value and denoted by **literal**(·).

One unique advantage of RDF graphs is that we are able to straightforwardly perform inference over the data represented in linked triples. By introducing RDF *schema*[5] (RDFS) or *ontology*, the meanings of predicates are defined, which then constrain how the data are interpreted to facilitate knowledge inference. For example, in RDFS, the pattern of the predicate *subClassOf* is:

```
IF
?A rdfs:subClassOf ?B
AND
?x rdf:type ?A .
THEN
?x rdf:type ?B .
```

In plain English, if one class in `A` is a subclass of another class `B` and `x` is an `A` (in the statement, `rdf:type` is a predicate describing the relationship *is-a*), then `x` is also a `B`. Thus, given an example graph as illustrated in Figure 2.1, which represents the knowledge of vegetarian diet, we can infer that, since *Jen* is a *Vegetarian*, which is a subclass of *Person*, *Jen* is a *Person*. Moreover, the same example also shows that we can use OWL[6] to compose ontology and achieve further

[5] https://www.w3.org/TR/rdf-schema/

[6] https://www.w3.org/TR/owl-features/

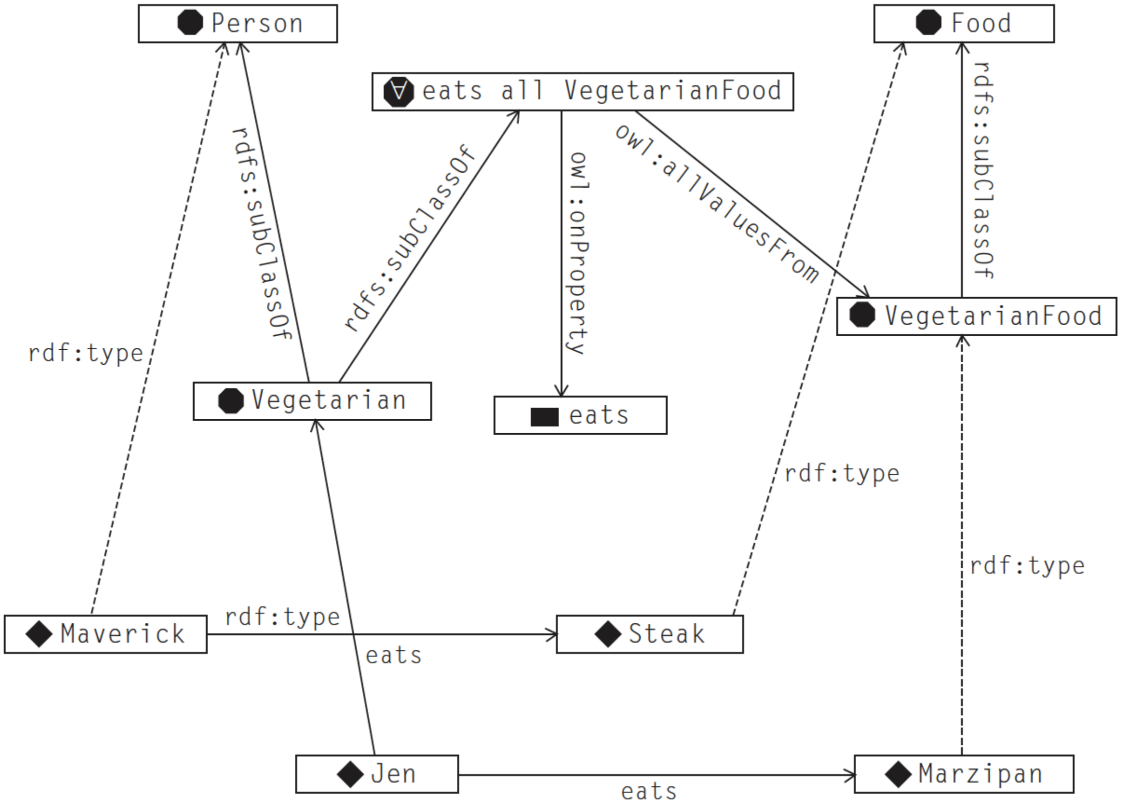


Figure 2.1: An example knowledge graph comprised of a collection of RDF triples. The figure is copied from Allemang & Hendler [1].

inference over the data. At the top of the graph, a node is defined by OWL (for simplicity, here we do not explain the details of OWL syntax) and imposes the restriction of *Vegetarian*: *eats* (all) *VegetarianFood*. Such a relationship lets us deduce that *Marzipan* is a *VegetarianFood*, because it is eaten by *Jen*, who is a *Vegetarian*. Note that this was not primitively asserted in the graph and denoted by a dotted arrow.

In 2012, *Google* announced its new application of the Semantic Web technology, which served as the complement to their main service, i.e., Web Search. The term *Knowledge Graph*[7] was then introduced as the project name. Thereafter, the success of Google's use of graph-based knowledge representation has drawn a lot of attention, and other Semantic Web projects also

[7] The term was mentioned in Google's blog post at https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

started advertising their technologies as knowledge graphs. Despite the popularity, there is still a lack of agreement on the definition of what a knowledge graph is. Rather, Paulheim [69] proposed a set of characteristics that are useful for distinguishing whether or not a collection of knowledge is a knowledge graph. They argued that a knowledge graph:

- mainly describes real-world entities and their interrelations, organized in a graph.
- defines possible classes and relations of entities in a schema.
- allows for potentially interrelating arbitrary entities with each other.
- covers various topical domains.

In the following, we present two representative knowledge graphs comprising our experimental datasets. Along with the general introduction, we also point out their characteristics in multilinguality, which is the core topic of this thesis.

### 2.1.1 DBpedia

DBpedia [50] is a popular knowledge graph, which is widely used in research projects as the test environment or the backbone of related applications. DBpedia has been publicly available since 2007, and its development is led by the Free University of Berlin and Leipzig University. Their aim is to extract the structured information in Wikipedia pages, e.g., *infoboxes*, on top of which they offer a standardized dataset represented as a knowledge graph. The construction process is aided by crowd-sourcing efforts and the ontology is collectively maintained by its user community. Due to its nature, DBpedia plays a role as the hub of knowledge graphs [28] and contains the most *owl:sameAs* predicates which are responsible for interlinking the entities referring to the same thing in different KGs. Up to April 2020, the entire DBpedia dataset consists of 21 billion RDF triples [41] and covers at least 125 languages.[8] It is worth mentioning that the infobox properties in different languages are mapped to the same one in DBpedia ontology, e.g., *author* and $\sigma\upsilon\gamma\gamma\rho\alpha\phi\epsilon\alpha\varepsilon$ (author in Greek) share the same identifier *dbo:author*, which facilitates data merge and augmentation across languages. To promote internationalization, the best practice for defining such multilingual infobox-to-ontology mappings is established alongside the original DBpedia Information Extraction Framework,[9] and the local communities also organize language-specific *chapters*[10] to support native developers and users.

[8] To the best of our knowledge, the latest official statistics regarding the number of languages is reported in https://wiki.dbpedia.org/about/facts-figures.

[9] https://wiki.dbpedia.org/documentation

[10] https://wiki.dbpedia.org/Internationalization/Chapters

### 2.1.2 Wikidata

Wikidata [99] is a collaboratively created knowledge graph started by Wikimedia Foundation[11] from 2012. Compared with DBpedia, Wikidata is more open source-centric and allows the contributors to continuously add and edit the information and schema. When users are creating or revising a statement (of a fact) on Wikidata, they are especially encouraged to provide references, which allow others to validate the asserted knowledge. This availability of information provenance improves the trustworthiness of Wikidata and is the main feature that sets it apart from other knowledge graphs. Another advantage of Wikidata is its proximity to Wikipedia, whose user base and experience helped Wikidata became popular rapidly since its launch, and the success also triggered Google's decision to close down its KG service, i.e., Freebase [9], and migrate the data to Wikidata. Recently, Tanon et al. [70] released the software to facilitate the migration and reported their ongoing efforts. Despite the challenges, when the merger is completed, Wikidata will then be considered the largest free knowledge graph in the world. One more remarkable characteristic of Wikidata is its unique design for multilingual data aggregation. In particular, a centralized graph is built primitively as the backbone, in which entities and properties are language-agnostic, and then information written in different languages are "attached" to such a unified graph, as opposed to the Wikipedia approach, i.e., each language is an independent edition.

## 2.2 Graph Embedding

The ubiquity of real-world phenomena that can be represented by graphs has motivated the use of graph analysis in many fields, e.g., protein-protein network analysis in biology [93] and friendship network analysis in sociology [31]. Similarly, graph-based methods have also been developed in natural language processing for many years, and one widely used example is word co-occurrence graphs [15]. Modeling textual information as graphs can bring new insights and enable useful applications from the perspective of entity-entity interactions, e.g., node clustering [68], node classification [102], and link prediction [51]. To drive such applications more effectively, graph embedding techniques are then proposed to convert graph features into a more compatible form for machine-learning solutions. Specifically, entities and relations are represented as dense vectors in a low-dimensional vector space that encode the information from the graphs. In these techniques, the mainstream ones are also machine learning-based and known collectively as graph representation learning.

[11] https://en.wikipedia.org/wiki/Wikimedia_Foundation

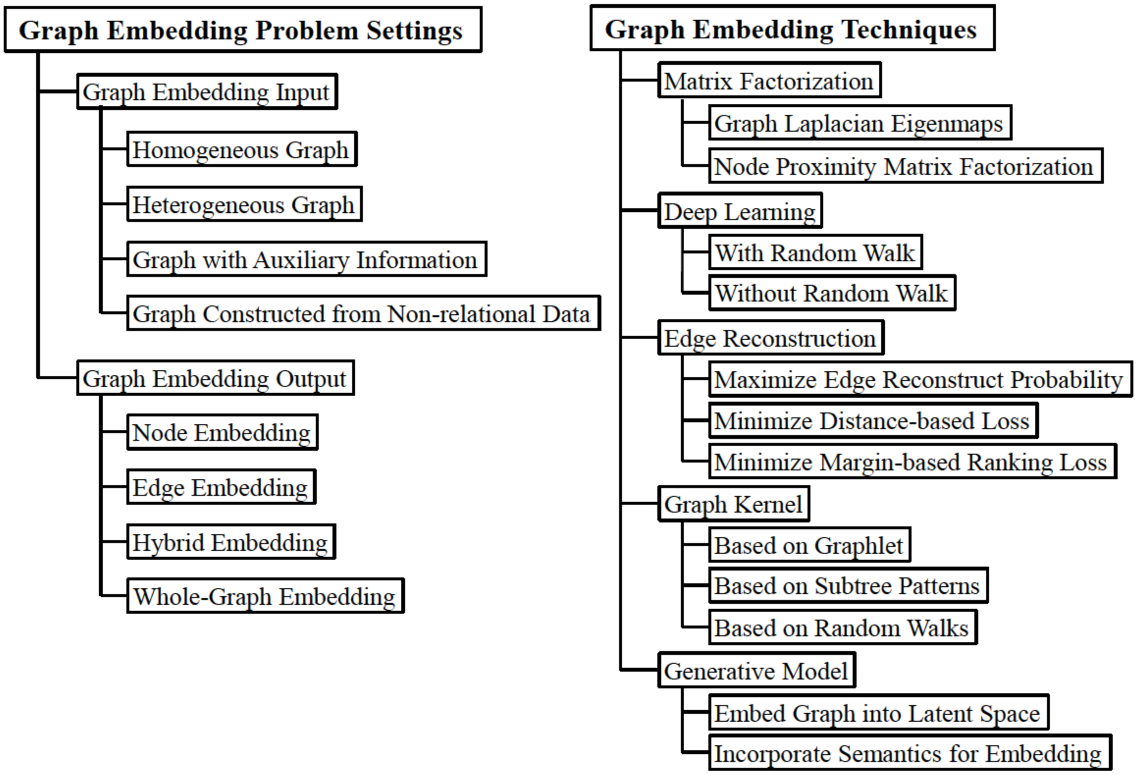

Graph Embedding Problem Settings
Graph Embedding Input
Homogeneous Graph
Heterogeneous Graph
Graph with Auxiliary Information
Graph Constructed from Non-relational Data
Graph Embedding Output
Node Embedding
Edge Embedding
Hybrid Embedding
Whole-Graph Embedding
Graph Embedding Techniques
Matrix Factorization
Graph Laplacian Eigenmaps
Node Proximity Matrix Factorization
Deep Learning
With Random Walk
Without Random Walk
Edge Reconstruction
Maximize Edge Reconstruct Probability
Minimize Distance-based Loss
Minimize Margin-based Ranking Loss
Graph Kernel
Based on Graphlet
Based on Subtree Patterns
Based on Random Walks
Generative Model
Embed Graph into Latent Space
Incorporate Semantics for Embedding

Figure 2.2: Taxonomy of graph embedding problems and techniques. The table is copied from Cai et al. [14].

Figure 2.2 shows the taxonomy of graph embedding techniques [14], out of which two categories are included in this thesis, namely edge reconstruction and deep learning. Most of our baseline methods use the former approach, and in contrast we explore the variants of the latter, particularly graph convolutional networks. We briefly introduce them in the following subsections:

### 2.2.1 Edge Reconstruction Approach

According to the design of objective function, edge reconstruction methods are mainly divided into two categories: (1) maximizing edge reconstruction probability (2) minimizing edge reconstruction loss, and (2) further branches out two subcategories, namely distance-based loss and margin-based ranking loss. In this section, we focus on the margin-based ranking loss, since all

| Model | Scoring Function |
|---|---|
| TransE [10] | $\|\|h + r - t\|\|_1$ |
| TransR [53] | $\|\|hM_r + r - tMr\|\|_2^2$ |
| TransH [103] | $\|\|(h - w_r^T hw_r) + d_r - (t - w_r^T tw_r)\|\|_2^2$ |
| DKRL [107] | $\|\|h_d + r - t_d\|\| + \|\|h_d + r - t_s\|\| + \|\|h_s + r - t_d\|\|$ |
| NTN [86] | $u_r^T \tanh(h^T W_r t + W_{rh} h + W_{rt} t + b_r)$ |

Table 2.1: Examples of edge reconstruction-based knowledge graph embedding models and the associated scoring functions.

the baseline models as well as the newly-proposed methods in this thesis adopt this objective.

Mathematically, given a knowledge graph $G$ which consists of a set of triples $(h, r, t)$, where $h$ ,$r$, and $t$ respectively denote head entity, relation, and tail entity, the margin-based ranking loss is defined as:

$$O = \min \sum_{(h,r,t)\in G} \sum_{(h',r,t')\in G'} \max\{0, \gamma + f_r(h, t) - f_r(h', t')\} \tag{2.1}$$

where $G'$ is the set of false triples, in which $h'$, $r$, and $t'$ also exist in $G$ but the triple as a whole, i.e., $(h', r, t')$, does not; $f_r(h, t)$ is a scoring function that evaluates the similarity between entities $h$ and $t$ with respect to relation $r$. One example of the scoring functions is $||h + r - t||_1$, which captures the translation relationship in the embedding space [10]. It is worth noting that the researchers of knowledge graph embedding largely dedicate to designing different scoring functions, and several representative ones are shown in Table 2.1.

### 2.2.2 Graph Convolutional Networks

Graph convolutional networks (GCNs) [46] are variants of convolutional neural networks, which have proven effective in capturing information from graph structures, such as dependency graphs [34], abstract meaning representation graphs [33], and knowledge graphs [105]. In practice, multi-layer GCNs are stacked to collect evidence from multi-hop neighbors. Formally, the $l$-th GCN layer takes as input feature representations $H^{(l-1)}$ and outputs $H^{(l)}$:

$$H^{(l)} = \phi\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l-1)}W^{(l)}\right) \tag{2.2}$$

where $\tilde{A} = A + I$ is the adjacency matrix, $I$ is the identity matrix, $\tilde{D}$ is the diagonal node degree matrix of $\tilde{A}$, $\phi(\cdot)$ is ReLU function, and $W^{(l)}$ represents learnable parameters in the $l$-th layer. $H^{(0)}$ is the initial input. GCNs can iteratively update the representation of each entity via a propagation mechanism through the graph.

To extend vanilla GCNs from non-relational graphs to knowledge graphs, Schlichtkrull et al. [84] proposed RGCN with relation-wise parameters for capturing relation information:

$$H^{(l)} = \phi\left(\sum_{r\in R}\hat{A}_r H^{(l-1)}W_r^{(l)}\right) \tag{2.3}$$

where $r \in R$ denotes the relation; $\hat{A}_r$ and $W_r^{(l)}$ are relation-specific normalized adjacency matrix and learnable parameters in $l$-th layers, respectively. Note that $\hat{A}_r$ can be normalized by node degrees or other pre-defined factors, and it is an identity matrix if $r$ indicates self-connection relation; otherwise its diagonal is zeros.

## 2.3 Pre-Trained Language Models

The effectiveness of pre-trained language models (PLMs) for natural language processing has been proven by substantial work. Such models are pre-trained with large corpora in a self-supervised fashion and the learned representation is universal to downstream tasks. After the fine-tuning stage, the performance on many of these tasks achieved the state of the art. Recently, the architectures of PLMs are further advanced from shallow to deep ones, which are capable of learning contextual word embeddings, in contrast to the previous generation, e.g., Word2vec [55] and Glove [71], which are context-free. BERT (Bidirectional Encoder Representation from Transformer) [25] and OpenAI GPT (Generative Pre-Training) [76] are the two most well-known deep pre-trained language models. The former emphasizes on text understanding and is regarded as an encoder model, and the latter rather specializes in text generation and is pre-trained by a decoding process. Both of them adopt Transformer [97] as their architecture, whilst the main difference is the design of attention masks. Specifically, the attention mechanism of BERT is global and bidirectional over all the tokens, but GPT only allows leftward and unidirectional attention for the purpose of autoregressive learning. Since we only exploit BERT in this work, the following will be focused on its details:

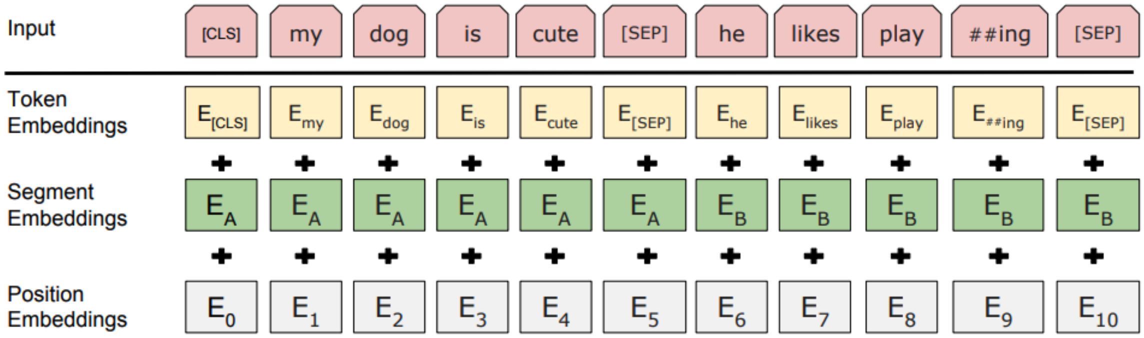


Figure 2.3: BERT input embeddings are comprised of three sets of embeddings, namely token, segment, and position embeddings. Segment embeddings indicate the sequence a token belongs to, and position embeddings represent the ordering information of tokens in a sentence. The illustration is copied from Delvin et al. [25].

## BERT

BERT is a Transformer-based language model, which computes token representations in parallel and is advantageous in modeling long-range dependencies. The token representations encode the contextual information bidirectionally, i.e., conditioning on both leftward and rightward context tokens in all layers. Similar to GPT, it also follows a two-stage framework: unsupervised pre-training and task-specific fine-tuning. Its input format is also carefully designed to be generic for downstream tasks. Specifically, a special token [CLS] is always placed at the beginning of an input sequence, and another special token [SEP] is used as a sentence-level delimiter when an input contains multiple sentences. The input embedding of each token consists of token, segment, and position embeddings, whose details are illustrated in Figure 2.3.

The main novelty of BERT resides in the proposed pre-training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In the MLM task, 15% of the input tokens are randomly masked and the objective is to recover the masked ones, i.e., predicting the original tokens by using the contextual information. Through this, the related context is embedded into the learned token representations. In NSP, two sentences are concatenated as the input and in 50% probability the second one actually follows the first as in the training corpus; the task objective is to predict whether the second sentence is "real" or randomly-picked. This task is devised for learning the relationships between sentences.

After BERT is pre-trained, we apply the knowledge and let it adapt to downstream tasks by fine-tuning with minimal architecture changes. In detail, task-specific layers (e.g., a multi-layer perceptron classifier) are appended after the output of BERT, and all the parameters are trained end-to-end with labeled data. When a task is formulated as a token-level prediction problem (e.g., named-entity recognition), the output representation of each token is fed into the aforementioned task-specific layers. As for sequence-level tasks (e.g., sentiment analysis), the output representation of [CLS] is employed as the sequence representation and serves as the input of the task-specific layers.

In addition to learning with English corpora, Multilingual BERT[12] (M-BERT), a variant of BERT pre-trained with multilingual Wikipedia text covering 104 languages, is also released, Interestingly, unlike traditional multilingual models, M-BERT are not explicitly pre-trained with parallel corpora. Nonetheless, the learned representations of M-BERT still yield strong performance on many downstream tasks [106]. To further understand M-BERT, Pires et al. [74] designed probing experiments and the results suggest that M-BERT is able to perform zero-shot cross-lingual transfer despite the low lexical overlap between source and target languages. They hypothesized that M-BERT benefits from the shared vocabulary set across all languages and the learned multilingual representations are capable of more than vocabulary memorization. However, recently Karthikeyan et al. [44] drew a contrary conclusion and discovered that when a bilingual BERT is pre-trained by English and a “fake” language,[13] which strictly has no vocabulary overlap with the target languages, it still performs comparably on the zero-shot task. Moreover, Artetxe et al. [3] showed that such transferability is achievable by even just swapping the vocabulary (i.e., the token embeddings) of a monolingual BERT, which further weakens the hypothesis.

## 2.4 Entity Matching

Research on knowledge graph alignment originated from the problem of entity matching in database community. MAGELLAN [48] is one of the representative works, which is an entity matching system equipped with a complete data pipeline, including blocking, matching, debugging, and sampling. Recently, deep learning-based techniques are also introduced to this task. For example, Ebraheem et al. [26] used recurrent neural networks to learn the embeddings and capture both syntactic and semantic similarities between entities in the vector space. Moreover, Mudgal et al. [60] explored the design space of neural network architectures for entity matching and divided the problem into three categories, namely structured, textual, and dirty. However, these systems only aim at relational data and the tables are assumed to be aligned in advance with

[12] https://github.com/google-research/bert/blob/master/multilingual.md

[13] The fake language is created by randomly shifting the characters in English Wikipedia text.

respect to the schema. Thus, such systems cannot be directly applied to aligning KGs, which are constructed using the RDF data model, instead of relational.

Out of the need to integrate heterogeneous knowledge graphs, the Semantic Web community has researched entity matching for KGs and held the evaluation event[14] for years, in which the entity matching is also known as instance matching. With respect to feature granularity, Castano et al. [16] categorize the matching techniques into value-oriented and record-oriented. The former defines the entity similarity based on the values of attributes, where most of the work focuses on textual values since string is the most commonly-used data type in knowledge graphs. By contrast, the latter considers more coarse-grained (record-level) features and contains four more subcategories, namely learning-based, similarity-based, rule-based, and context-based.

Language-wise, such an alignment problem can be separated into monolingual and cross-lingual entity matching. For the monolingual problem, the main approaches are to match two entities by computing string similarity of entity labels [83, 98, 62] or topological similarity of graph structures [77, 72, 37, 5]. Additionally, Trsedya et al. [96] proposed an alignment framework that also incorporates attribute values to learn the entity embeddings. To match entities across different languages, recent studies [18, 92] learned cross-lingual entity embeddings based on TransE [10]. Chen et al. [17] further proposed a co-training algorithm to incorporate multilingual textual information by alternately learning entity and description embeddings. Moreover, Wang et al. [105] applied GCNs with the connectivity matrix defined on relations to embed entities from multilingual KGs into a unified low-dimensional space.

[14] http://oaei.ontologymatching.org/

# Chapter 3

# Proposed Approach

Existing KGs [8, 91, 80] provide multi-aspect information of entities. The key challenge is how to utilize the provided features to learn better embeddings of entities. In this section, we introduce four neural models that incorporate different aspects of information as well as the learning objective for cross-lingual entity matching. The proposed models are mainly divided into two modules, namely GCN-based and BERT-based. The GCNs are primarily used to process the topological, relation, and attribution features, and we further propose a hybrid variant that integrates feedforward neural networks to control the noise from graph propagation. To incorporate literal descriptions, we apply the BERT-based models and additionally devise a novel training scheme which can largely reduce the time complexity while running inference. Finally, we introduce two integration strategies to combine these two modules.

## 3.1 Multi-Aspect Alignment Networks

GCNs can iteratively update the representation of each entity node via a propagation mechanism through the graph. Inspired by previous studies [110, 105], we also adopt GCNs in this work to collect evidence from multilingual KG structures and to learn cross-lingual embeddings of entities. The primary assumptions are: (1) equivalent entities tend to be neighbored by equivalent entities via the same types of relations; (2) equivalent entities tend to share similar or even the same attributes. In the following, we first discuss how we construct raw features for the three aspects, which are then fed as inputs to our models, and use $X_t$, $X_r$, and $X_a$ to denote the topological, relation, and attribute features, respectively.

The topological features are designed to reflect neighborhood proximity information of entities, which can be captured by multi-layer GCNs. Following Wang et al. [105], we set the initial topological features to $X_t = I$, i.e., an identity matrix serving as index vectors, so that the GCNs learn the topological embeddings. In addition, we also consider the relation and attribute features. As shown in Figure 1.1, the connected relations and attributes of two equivalent entities, e.g., "*University of Toronto*" (English) and "トロント大学" (Japanese), have a lot of overlap, which can benefit cross-lingual entity matching. Specifically, they share the same relation types, e.g., "country" and "almaMater", and some attributes, e.g., "foundDate" and "創立年". To capture relation information, Schlichtkrull et al. [84] proposed RGCN with relation-wise parameters. However, with regard to this task, existing KGs typically contain thousands of relation types but few pre-aligned ILLs. Directly applying RGCN may introduce too many parameters for the limited training data and thus cause overfitting. Wang et al. [105] instead simply used the unlabeled GCNs [46] with two proposed measures (i.e., functionality and inverse functionality) to encode the information of relations into the adjacency matrix. They also considered attributes as input features in their architecture. However, this approach may lose information about relation types. Therefore, we regard relations and attributes of entities as bag-of-words features to explicitly model these two aspects. Specifically, we construct count-based *N-hot* vectors $X_r$ and $X_a$ for these two aspects of features, respectively, where the $(i, j)$ entry is the count of the $j$-th relation (attribute) for the corresponding entity $e_i$. Note that we only consider the top-$F$ most frequent relations and attributes to avoid data sparsity issues. Thus, for each entity, both of its relation and attribute features are $F$-dimensional vectors.

With the features constructed as above, we propose the *Multi-Aspect Alignment Network* (MAN) to leverage the three aspects of information. Specifically, three $l$-layer GCNs take as inputs the three aspects of features (i.e., $X_t$, $X_r$, and $X_a$) and produce the representations $H_t^{(l)}$, $H_r^{(l)}$, and $H_a^{(l)}$ according to Equation 2.2. Finally, the multi-aspect entity embedding is:

$$H_m = [H_t^{(l)} \oplus H_a^{(l)} \oplus H_r^{(l)}] \tag{3.1}$$

where $\oplus$ denotes vector concatenation. $H_m$ can then be used for making alignment decisions.

Such fusion through concatenation is also known as *Scoring Level Fusion*, which has been proven simple but effective for capturing multi-modal semantics [13, 45, 21]. It is worth noting that the main differences between MAN and the work of Wang et al. [105] are twofold: (1) we use the same approach as Kipf & Welling [46] to construct the adjacency matrix, while Wang et al. [105] designed a new connectivity matrix as the adjacency matrix for the GCNs; (2) MAN explicitly regards the relation type features as model input, while Wang et al. [105] incorporated such relation information into the connectivity matrix.

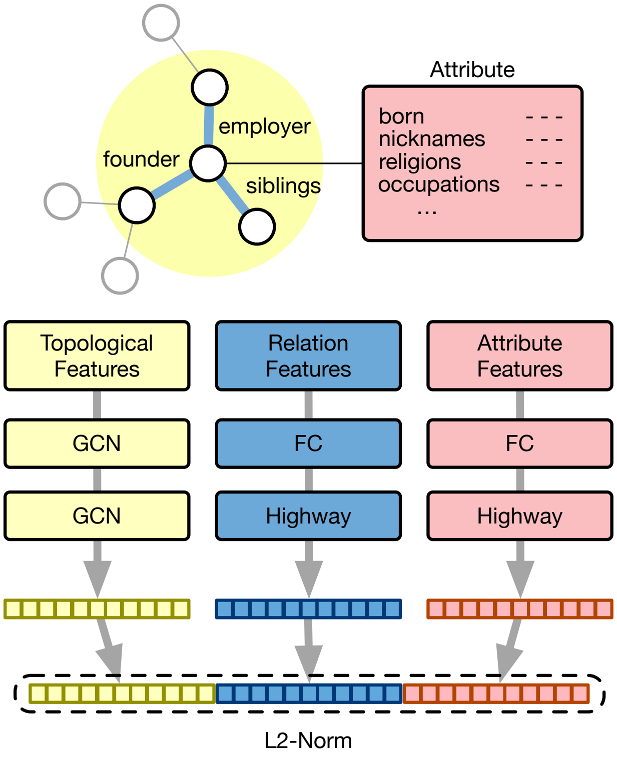


Figure 3.1: Architecture overview of HMAN, where the GCNs are employed to analyze topological features and the feedforward networks are responsible for relation and attribute features. The three aspects are fused by concatenation, and L2 normalization is applied to the final representation. (The architecture of MAN can be derived by replacing the FC and highway layers with GCN layers.)

## 3.2 Hybrid Multi-Aspect Alignment Networks

Note that MAN propagates relation and attribute information through the graph structure. However, for aligning a pair of entities, we observe that considering the relations and attributes of neighboring entities, besides their own ones, may introduce noise. Merely focusing on relation and attribute features of the current entity could be a better choice. Thus, we propose the *Hybrid Multi-Aspect Alignment Network* (HMAN) to better model such diverse features, shown in Figure 3.1. Similar to MAN, we still adopt the $l$-th layer of a GCN to obtain topological embeddings $H_t^{(l)}$, but exploit feedforward neural networks to obtain the embeddings of relations and attributes. The feedforward neural networks consist of one fully-connected (FC) layer and a highway network layer [89]. The reason we use highway networks is consistent with the conclusions of Mudgal et al. [60], who conducted a design space exploration of neural models for entity matching and found that highway networks are generally better than FC layers in convergence speed and effectiveness.

Formally, these feedforward neural networks are defined as:

$$\begin{aligned} S_f &= \phi(W_f^{(1)} X_f + b_f^{(1)}) \\ T_f &= \sigma(W_f^t S_f + b_f^t) \\ G_f &= \phi(W_f^{(2)} S_f + b_f^{(2)}) \cdot T_f + S_f \cdot (1 - T_f) \end{aligned} \tag{3.2}$$

where $f \in \{r, a\}$ and $X_f$ refer to one specific aspect (i.e., relation or attribute) and the corresponding raw features, respectively, $W_f^{(1,2,t)}$ and $b_f^{(1,2,t)}$ are model parameters, $\phi(\cdot)$ is ReLU function, and $\sigma(\cdot)$ is sigmoid function. Accordingly, we obtain the hybrid multi-aspect entity embedding $H_y = [H_t^{(l)} \oplus G_r \oplus G_a]$, to which $\ell_2$ normalization is further applied. It is worth noting that we can also derive the illustration of MAN by replacing the FC and highway layers in Figure 3.1 with GCN layers.

## 3.3 PointwiseBERT

Existing multilingual KGs [8, 61, 80] also provide literal descriptions of entities expressed in different languages and contain detailed semantic information about the entities. The key observation is that literal descriptions of equivalent entities are semantically close to each other. However, it is non-trivial to directly measure the semantic relatedness of two entities' descriptions, since they are expressed in different languages.

As mentioned in Section 2.3, BERT [25] has advanced the state-of-the-art in various NLP tasks and its multilingual variant, i.e., M-BERT, also yields strong performance. The spirit of M-BERT in the multilingual scenario is to project words or sentences from different languages into the same semantic space. This aligns well with our objective — bridging gaps between descriptions written in different languages. Therefore, we include M-BERT into our framework for cross-lingual entity matching.

The most straightforward BERT-based approach is to formulate our task as a text matching task. For two entities $e_1$ and $e_2$ from two KGs in $L_1$ and $L_2$, denoting source language and target language, respectively, their textual descriptions are $d_1$ and $d_2$, consisting of word sequences in two languages. The model takes as inputs [CLS] $d_1$ [SEP] $d_2$ [SEP], where [CLS] is the special classification token, from which the final hidden state is used as the sequence representation,

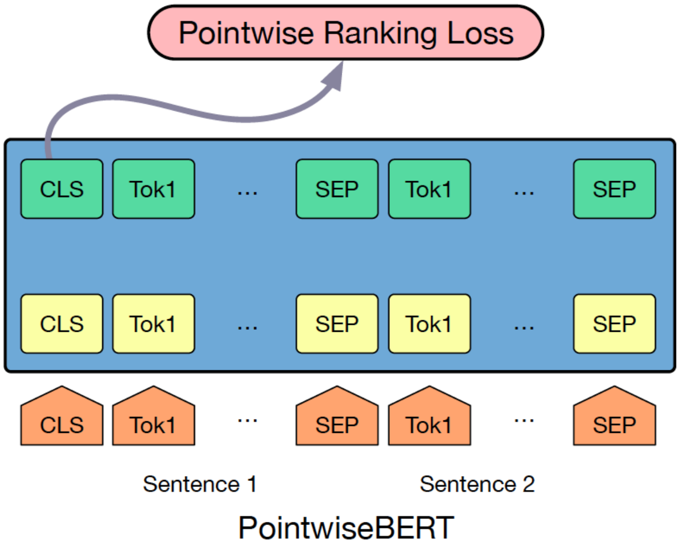


Figure 3.2: Architecture overview of POINTWISEBERT, which is basically the vanilla approach of BERT for classifying a pair of sentences. Cross entropy is applied as the loss.

and [SEP] is the special token for separating token sequences, and produces the probability of classifying the pair as equivalent entities. The probability is then used to rank all candidate entity pairs, i.e., ranking score. We denote this vanilla approach as POINTWISEBERT, shown in Figure 3.2, in contrast to the novel method proposed in Section 3.4, i.e., PAIRWISEBERT.

Nonetheless, this approach is computationally expensive, since for each entity we need to consider all candidate entities in the target language, and under this scheme *full ranking* is practically intractable because of the significant cost of BERT inference. One feasible solution, inspired by Shi et al. [85], is to reduce the search space for each entity with a *reranking strategy* (see Section 3.6).

## 3.4 PairwiseBERT

Due to the heavy computational cost of POINTWISEBERT, semantic matching between all entity pairs is very expensive. Instead of producing ranking scores for description pairs, we propose PAIRWISEBERT to encode the entity literal descriptions as cross-lingual textual embeddings, where distances between entity pairs can be directly measured using these embeddings.

The PAIRWISEBERT model consists of two components, each of which takes as input the

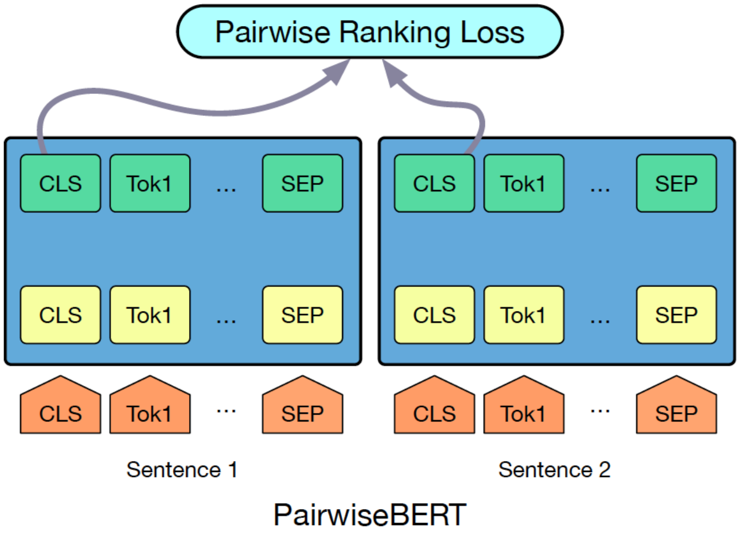


Figure 3.3: Architecture overview of PairwiseBert, where a BERT is reused for both source and target sentences and Equation 3.3 is applied as the loss.

description of one entity (from the source or target language), as depicted in Figure 3.3. Specifically, the input is designed as [CLS] $d_1(d_2)$ [SEP], which is then fed into PairwiseBert for contextual encoding. We select the hidden state of [CLS] as the textual embedding of the entity description for training and inference.

## 3.5 Model Objective

To address our task, we follow the edge reconstruction approach mentioned in Section 2.2 to learn the entity embeddings for entity matching. During the training phase, the goal is to embed cross-lingual entities into the same low-dimensional vector space where equivalent entities are close to each other. Mathematically, given two knowledge graphs, $\mathcal{G}_1$ and $\mathcal{G}_2$, and a set of pre-aligned entity pairs $I\left(\mathcal{G}_1, \mathcal{G}_2\right)$ as training data, our model is trained to minimize the margin-based ranking loss defined as:

$$J = \sum_{(e_1,e_2)\in I} \sum_{(e_1',e_2')\in I'} [\rho(h_{e_1}, h_{e_2}) + \beta - \rho(h_{e_1'}, h_{e_2'})]_+ \tag{3.3}$$

where $[x]_+ = \max\{0, x\}$, $h_e$ is the vector representation of entity $e$, $I'$ denotes the set of negative entity alignment pairs constructed by corrupting the gold pair $(e_1, e_2) \in I$. In specific, we replace

$e_1$ or $e_2$ with a randomly-chosen entity in $E_1$ or $E_2$. $\rho(x, y)$ is the $\ell_1$ distance function, and $\beta > 0$ is the margin hyperparameter separating positive and negative pairs.

Note that this objective function is not applied to POINTWISEBERT, which is a standard classifier and outputs a similarity score of an entity pair, as opposed to entity embeddings. We just follow Devlin et al. [25] and optimize this model with cross entropy loss.

## 3.6 Integration Strategies

Up to here, we have introduced the two modules that separately collect evidence from knowledge graph structures and the literal descriptions of entities, based on GCNs and M-BERT, respectively. In this section, we present two strategies to integrate these two modules to further boost performance:

### Reranking

As mentioned in Section 3.3, the POINTWISEBERT model takes as input the concatenation of two descriptions for each candidate–entity pair, where conceptually we must process every possible pair in the training set. Such a setting would lead to a prohibitive computational cost

One way to reduce the cost of POINTWISEBERT is to ignore candidate pairs that are unlikely to be aligned. Rao et al. [79] showed that uncertainty-based sampling can provide extra improvements in ranking. Following this idea, the GCN-based models (i.e., MAN and HMAN) are used to generate a candidate pool whose size is much smaller than the entire universe of entities. Specifically, GCN-based models provide top-$q$ candidates of target entities for each source entity (where $q$ is a hyperparameter). Then, the POINTWISEBERT model produces a ranking score for each candidate–entity pair in the pool to further rerank the candidates. However, the weakness of such a reranking strategy is that performance is bounded by the quality of (potentially limited) candidates produced by MAN or HMAN.

### Weighted Concatenation

With the textual embeddings learned by PAIRWISEBERT denoted as $H^B$ and graph embeddings denoted as $H^G$, a simple way to combine the two modules is by weighted concatenation:

$$H^C = \tau \cdot H^G \oplus (1 - \tau) \cdot H^B \tag{3.4}$$

where $H^G$ is the graph embeddings learned by either MAN or HMAN, and $\tau$ is a factor to balance the contribution of each source (where $\tau$ is a hyperparameter).

## 3.7 Ranking-Based Entity Matching

After we obtain the embeddings of entities, we leverage $\ell_1$ distance to measure the distance between candidate–entity pairs. A small distance reflects a high probability for an entity pair to be aligned as equivalent entities. To implement the reranking strategy, we select the target entities that have the smallest distances to a source entity in the vector space learned by MAN or HMAN as its candidates. For weighted concatenation, we employ the $\ell_1$ distance of the representations of a pair derived from the concatenated embedding, i.e., $H^C$, as the ranking score.

# Chapter 4

# Experimental Results

## 4.1 Datasets and Settings

To evaluate our methods, we use the same benchmark datasets as in the previous work [92, 105], namely DBP15K and DBP100K. In addition, we further create a new dataset, referred to as XEM15, to simulate the extensive real-world challenges in cross-lingual entity matching. The details are as follows:

### 4.1.1 DBP15K and DBP100K

Table 4.1 outlines the statistics of DBP15K and DBP100K datasets, which contain 15,000 and 100,000 ILLs, respectively. Both are divided into three subsets: Chinese-English (ZH-EN), Japanese-English (JA-EN), and French-English (FR-EN). Following Wang et al. [105], we adopt the same split settings, where 30% of the ILLs are used as training and the remaining 70% for evaluation.

In all our experiments, we employ two-layer GCNs and the top 1000 (i.e., $F$=1000) most frequent relation types and attributes are included to build the $N$-hot feature vectors. For the MAN model, we set the dimensionality of topological, relation, and attribute embeddings to 200, 100, and 100, respectively. When training HMAN, the hyperparameters are dependent on the dataset sizes due to GPU memory limitations. For DBP15K, we set the dimensionality of topological embeddings, relation embeddings, and attribute embeddings to 200, 100, and 100, respectively. For DBP100K, the dimensionalities are set to 100, 50, and 50, respectively. We adopt SGD to update parameters and the numbers of epochs are set to 2,000 and 50,000 for MAN and HMAN,

| Datasets | | DBP15K | | | | |
|---|---|---|---|---|---|---|
| | | Entities | Rel. | Attr. | Rel.triples | Attr.triples |
| ZH-EN | Chinese | 66,469 | 2,830 | 8,113 | 153,929 | 379,684 |
| | English | 98,125 | 2,317 | 7,173 | 237,674 | 567,755 |
| JA-EN | Japanese | 65,744 | 2,043 | 5,882 | 164,373 | 354,619 |
| | English | 95,680 | 2,096 | 6,066 | 233,319 | 497,230 |
| FR-EN | French | 66,858 | 1,379 | 4,547 | 192,191 | 528,665 |
| | English | 105,889 | 2,209 | 6,422 | 278,590 | 576,543 |
| Datasets | | DBP100K | | | | |
| | | Entities | Rel. | Attr. | Rel.triples | Attr.triples |
| ZH-EN | Chinese | 106,517 | 4,431 | 16,152 | 329,890 | 1,404,615 |
| | English | 185,022 | 3,519 | 14,459 | 453,248 | 1,902,725 |
| JA-EN | Japanese | 117,836 | 2,888 | 12,305 | 413,558 | 1,474,721 |
| | English | 118,570 | 2,631 | 13,238 | 494,087 | 1,738,803 |
| FR-EN | French | 105,724 | 1,775 | 8,029 | 409,399 | 1,361,509 |
| | English | 107,231 | 2,504 | 13,170 | 513,382 | 1,957,813 |

Table 4.1: Statistics of DBP15K and DBP100K. Rel. and Attr. stand for relations and attributes, respectively.

respectively. The margin $\beta$ in the loss function is set to 3. The balance factor $\tau$ is determined by grid search, which shows that the best performance lies in the range from 0.8 to 0.7. For simplicity, $\tau$ is set to 0.8 in all associated experiments. Multilingual BERT-base models with 768 hidden units are used in PointwiseBert and PairwiseBert. We additionally append one more FC layer to the representation of [CLS] and reduce the dimensionality to 300. Both BERT models are fine-tuned using Adam optimizer.

### 4.1.2 XEM15

The current datasets for cross-lingual entity matching tend to ignore the situation that, given a source entity, the counterpart in the target language may not even exist, which is fairly common in the real-world practice. Such datasets essentially assumed a nearly perfect world that all the entities are already built in the multilingual KGs and merely part of ILLs are missing. However, this assumption would hinder the development of related techniques, particularly the capability of predicting the absence of correspondence, also known as NIL (not-in-list) prediction. Also, the main motivation of cross-lingual entity matching is to augment the contents for low-resource languages by leveraging resource-rich ones. Therefore, testing the effectiveness under the settings of low-resource languages is important. However, most of the adopted languages in the previous datasets are widely spoken, e.g., French, Japanese, and Chinese. As a result, the associated KGs of such languages are relatively complete and barely reflect the realistic difficulty of low-resource languages, e.g., Tamil or Kannada.

To draw closer to the real-world scenario, we introduce a new benchmark dataset featured with unlinkable cases and 15 low-resource languages for cross-lingual entity matching, which we refer to as XEM15. In our definition, a language is regarded as low-resource if it is ranked lower than 25 by the number of Wikipedia articles.[1] We carefully selected the languages into XEM15 to preserve the diversity. Specifically, we first singled out three groups of languages, namely rank 26-50, 51-75, and 76-113, from each of which we then manually choose 5 languages (i.e., 15 in total) to ensure every language uses a unique script, e.g., if Farsi is chosen, no more languages written in Persian script are allowed. To produce the testbed KGs and pre-aligned ILLs, we randomly sample 15K seed entities for each language from its Wikidata repository (September 2020). Only the entities that have Wikipedia sitelinks are considered as existing, and ILLs are extracted if there are corresponding sitelinks in English; otherwise, the samples are labeled as unlinkable cases. In other words, each low-resource language has 15K labeled examples, part of which targets are NIL. Next, we collect their attributes and neighboring entities along with the relations,[2] including both outward and inward ones, to create the monolingual KGs. Moreover, for simplicity, we filter out the entities created to organize Wikipedia contents, e.g., Wikimedia category or template. The statistics of all these languages and the associated KGs are listed in Table 4.2.

Note that the way we label unlinkable cases inevitably runs the risk of false-negative error, i.e., an English counterpart is actually existing in Wikidata but not recognized. If these entities are included in our dataset but the ground truth is wrongly specified as NIL, the evaluation will be inaccurate. Therefore, from the standpoint of dataset creation, we adopt English as the unified target language of XEM15, because it is the most co-edited language for multilingual Wikipedians [35] and is least likely that an existing entity is overlooked. To empirically assess the quality, we also manually verified an adequate number of NIL cases; the keywords in entity descriptions were translated to search for the corresponding pages on English Wikipedia. Due to the limitation in the resource of human translators, we only inspected the Cantonese KG, in which 94[3] random samples out of the 3,820 NIL cases were studied. As a result, one false negative sample was found, i.e., *Q55719486*, whose English sitelink resides in *Q85521565* on Wikidata. However, we further confirmed that this English counterpart (i.e., *Q85521565*) does not exist in XEM15, which means in our simulated task it is still proper to regard this case as

[1] https://en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics

[2] In Wikidata, the predicates are called *properties* and we regard the ones whose objects are entities as relations and the others as attributes.

[3] The worst-case 95% confidence interval is ±10%.

NIL. In the future, we will seek more resources and verify the other languages to better ensure the quality of XEM15.

To facilitate comparison across different languages, we set the matching task to be unidirectional, i.e., mapping from the 15 low-resource languages to English, and distinctively construct our English KG. Specifically, all the English counterparts of the seeds sampled from the 15 low-resource languages, i.e., the union set, are collectively configured as the English seed entities. Then, the same approach is used to build the graph, i.e., inserting the one-hop relation/attribute triples. In contrast to the previous datasets, where the monolingual KGs are not reused in different tasks of language pairs (e.g., Japanese-English and French-English are both evaluated but each time the English KG is resampled), our English KG is shared and better serves as a standardized reference.

In this thesis, we adopt the state-of-the-art models on DBP15K and DBP100K, i.e., HMAN and PAIRWISEBERT, to further evaluate on XEM15 and investigate how different levels of data scarcity in low-resource languages affect the models. We follow the same data split setting as in DBP15K (i.e., train-dev-test ratio = 27:3:70), but the dev set is kept for the use of validation, as opposed to being merged into the training set [105]. As for the model settings, since the scale of KGs in XEM15 is similar to DBP100K, we apply most of the same hyperparameters to HMAN and PAIRWISEBERT except for three: first, as there are relatively fewer relation types and attributes in this dataset, we only extract the top 300 as the features; Second, we also found that HMAN converges faster in the preliminary tests, so the number of epochs is decreased from 50,000 to 10,000; Third, according to the optimal results from grid search, we adjust the balance factor $\tau$ to 0.7.

## 4.2 Evaluation Metric

Following the previous work [10, 92, 105], *Hits@k* is used as the evaluation metric, which measures the proportion of correctly aligned entities ranked in the top-$k$ candidates. The results of DBP15K and DBP100K are reported in both directions, e.g., ZH-EN and EN-ZH.

## 4.3 Results

### 4.3.1 Comparison with Other Models

In this section, we investigate whether the proposed GCN-based and BERT-based modules can efficiently leverage different aspects of information in multilingual KGs. DBP15K and DBP100K

| XEM15 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Language | Abbr. | Entities | Rel. | Attr. | Rel.triples | Attr.triples | NIL% |
| English | EN | 415,264 | 927 | 4,786 | 1,705,731 | 2,088,482 | - |
| Farsi | FA | 43,030 | 455 | 1,099 | 101,051 | 148,371 | 14.0 |
| Korean | KO | 40,138 | 493 | 1,116 | 91,867 | 117,161 | 35.2 |
| Armenian | HY | 36,870 | 276 | 175 | 102,107 | 93,892 | 38.4 |
| Hebrew | HE | 36,541 | 485 | 1,005 | 102,713 | 176,389 | 21.7 |
| Greek | EL | 35,470 | 413 | 640 | 107,580 | 134,868 | 25.5 |
| Georgian | KA | 30,277 | 307 | 225 | 85,362 | 98,717 | 26.5 |
| Cantonese | YUE | 29,895 | 208 | 81 | 80,185 | 55,345 | 25.5 |
| Urdu | UR | 28,901 | 267 | 264 | 78,569 | 114,053 | 16.2 |
| Macedonian | MK | 28,606 | 470 | 1,860 | 91,671 | 197,364 | 30.3 |
| Bengali | BN | 24,916 | 343 | 716 | 77,973 | 123,245 | 17.0 |
| Hindi | HI | 22,621 | 193 | 82 | 46,032 | 38,388 | 44.9 |
| Tamil | TA | 22,244 | 245 | 133 | 52,061 | 38,068 | 37.7 |
| Malayalam | ML | 21,522 | 250 | 133 | 59,961 | 81,761 | 22.4 |
| Tagalog | TL | 20,950 | 131 | 30 | 51,703 | 29,511 | 12.9 |
| Kannada | KN | 16,672 | 170 | 56 | 33,708 | 41,500 | 35.3 |

Table 4.2: Statistics of XEM15. Rel. and Attr. stand for relations and attributes, respectively; NIL% indicates the percentage of absent English counterparts.

are employed as the benchmark datasets to compare with the baseline methods.

## Results on Graph Embeddings

We first compare Man and Hman against previous systems, namely JE [37], MTransE [19], JAPE [92], and GCN [105]. As shown in Table 4.3, Man and Hman consistently outperform all baselines in all scenarios, especially Hman. It is worth noting that, in this case, Man and Hman use the same amount of information as the GCN [105], while JAPE [92] requires extra supervised labels (relations and attributes of two KGs need to be aligned in advance). The performance improvements confirm that our model can better utilize topological, relational, and attribute information of entities provided by KGs.

Moreover, we perform ablation studies on the two proposed models to investigate the effectiveness of each component. We alternatively remove each aspect of features (i.e., topological, relation, and attribute features) and the highway layer in Hman, denoted as w/o TE (RE, AE, and HW). As reported in Table 4.3, we observe that after removing relation or attribute features, the performance of Hman and Man drops across all datasets. These figures prove that these two aspects of features are useful in making alignment decisions. On the other hand, compared with

| Model | ZH → EN | | | EN→ ZH | | | JA → EN | | | EN→ JA | | | FR → EN | | | EN→ FR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 |
| | DBP15K | | | | | | | | | | | | | | | | | |
| JE | 21.2 | 42.7 | 56.7 | 19.5 | 39.3 | 53.2 | 18.9 | 39.9 | 54.2 | 17.8 | 38.4 | 52.4 | 15.3 | 38.8 | 56.5 | 14.6 | 37.2 | 54.0 |
| MTransE | 30.8 | 61.4 | 79.1 | 24.7 | 52.4 | 70.4 | 27.8 | 57.4 | 75.9 | 23.7 | 49.9 | 67.9 | 24.4 | 55.5 | 74.4 | 21.2 | 50.6 | 69.9 |
| JAPE | 41.1 | 74.4 | 88.9 | 40.1 | 71.0 | 86.1 | 36.2 | 68.5 | 85.3 | 38.3 | 67.2 | 82.6 | 32.3 | 66.6 | 83.1 | 32.9 | 65.9 | 82.3 |
| GCN | 41.2 | 74.3 | 86.2 | 36.4 | 69.9 | 82.4 | 39.9 | 74.4 | 86.1 | 38.4 | 71.8 | 83.7 | 37.2 | 74.4 | 86.7 | 36.7 | 73.0 | 86.3 |
| MAN | 46.0 | 79.4 | 90.0 | 41.5 | 75.6 | 88.3 | 44.6 | 78.8 | 90.0 | 43.0 | 77.1 | 88.7 | 43.1 | 79.7 | 91.7 | 42.1 | 79.1 | 90.9 |
| MAN w/o TE | 21.5 | 55.0 | 79.4 | 20.2 | 53.6 | 78.8 | 15.0 | 44.0 | 69.9 | 14.3 | 44.0 | 70.6 | 10.2 | 34.5 | 59.5 | 10.8 | 35.2 | 60.3 |
| MAN w/o RE | 45.6 | 79.1 | 89.5 | 41.1 | 75.0 | 87.3 | 44.2 | 78.7 | 89.8 | 43.0 | 76.9 | 88.1 | 42.8 | 79.7 | 91.4 | 42.1 | 78.9 | 90.6 |
| MAN w/o AE | 43.7 | 77.1 | 87.8 | 39.2 | 72.9 | 85.5 | 43.2 | 77.6 | 88.4 | 41.2 | 74.9 | 86.6 | 42.9 | 79.6 | 91.0 | 41.5 | 78.9 | 90.5 |
| HMAN | **56.2** | **85.1** | **93.4** | **53.7** | **83.4** | **92.5** | **56.7** | **86.9** | **94.5** | **56.5** | **86.6** | **94.6** | **54.0** | **87.1** | **95.0** | **54.3** | **86.7** | **95.1** |
| HMAN w/o TE | 3.2 | 16.7 | 38.3 | 3.5 | 17.2 | 38.5 | 5.4 | 22.3 | 45.5 | 5.2 | 22.0 | 45.5 | 2.4 | 13.9 | 35.3 | 2.2 | 13.7 | 35.3 |
| HMAN w/o RE | 50.2 | 78.4 | 86.5 | 49.3 | 78.6 | 87.0 | 52.6 | 81.6 | 89.1 | 52.4 | 81.1 | 89.8 | 52.7 | 84.2 | 91.4 | 52.0 | 83.9 | 91.1 |
| HMAN w/o AE | 49.2 | 81.0 | 89.8 | 48.8 | 80.9 | 90.0 | 52.2 | 83.3 | 91.6 | 51.5 | 83.1 | 91.6 | 52.3 | 85.6 | 93.7 | 52.3 | 85.1 | 93.2 |
| HMAN w/o HW | 46.8 | 76.1 | 84.1 | 46.0 | 76.2 | 84.6 | 50.5 | 79.5 | 87.5 | 49.9 | 79.1 | 87.5 | 51.9 | 82.7 | 90.9 | 51.6 | 82.5 | 90.6 |
| | DBP100K | | | | | | | | | | | | | | | | | |
| JE | - | 16.9 | - | - | 16.6 | - | - | 21.1 | - | - | 20.9 | - | - | 22.9 | - | - | 22.6 | - |
| MTransE | - | 34.3 | - | - | 29.1 | - | - | 33.9 | - | - | 27.2 | - | - | 44.8 | - | - | 39.1 | - |
| JAPE | 20.2 | 41.2 | 58.3 | 19.6 | 39.4 | 56.0 | 19.4 | 42.1 | 60.5 | 19.1 | 39.4 | 55.9 | 26.2 | 54.6 | 70.5 | 25.9 | 51.3 | 66.9 |
| GCN | 23.1 | 47.5 | 63.8 | 19.2 | 40.3 | 55.4 | 26.4 | 55.1 | 70.0 | 21.9 | 44.4 | 56.6 | 29.2 | 58.4 | 68.7 | 25.7 | 50.5 | 59.8 |
| MAN | 27.2 | 54.2 | **72.8** | 24.7 | 50.2 | **69.0** | 30.0 | 60.4 | **77.3** | 26.6 | 54.4 | 71.2 | 31.6 | 64.0 | 77.3 | 28.8 | 59.3 | 73.4 |
| MAN w/o TE | 11.8 | 28.6 | 47.7 | 11.2 | 28.3 | 47.9 | 7.4 | 21.7 | 39.4 | 7.2 | 21.6 | 39.8 | 5.4 | 19.4 | 38.2 | 5.1 | 18.8 | 37.1 |
| MAN w/o RE | 26.5 | 53.4 | 72.1 | 23.9 | 49.2 | 67.9 | 29.8 | 60.3 | 77.1 | 26.3 | 53.9 | 70.6 | 31.0 | 63.2 | 76.4 | 28.4 | 58.4 | 72.2 |
| MAN w/o AE | 25.5 | 51.7 | 70.4 | 22.8 | 47.6 | 66.3 | 29.4 | 59.4 | 76.1 | 25.9 | 52.9 | 69.7 | 30.8 | 62.7 | 75.8 | 28.1 | 57.8 | 71.5 |
| HMAN | **29.8** | **54.6** | 69.5 | **28.7** | **53.3** | **69.0** | **34.3** | **63.3** | 76.1 | **33.8** | **63.0** | **76.7** | **37.5** | **67.7** | **77.7** | **37.6** | **68.1** | **78.5** |
| HMAN w/o TE | 6.8 | 20.3 | 39.2 | 7.2 | 21.0 | 39.4 | 3.0 | 11.5 | 27.3 | 3.3 | 11.8 | 28.0 | 0.5 | 3.5 | 11.1 | 0.5 | 3.4 | 11.4 |
| HMAN w/o RE | 28.0 | 50.3 | 62.3 | 28.2 | 50.6 | 62.9 | 30.3 | 54.9 | 64.8 | 30.2 | 55.9 | 66.9 | 32.8 | 60.3 | 69.1 | 33.3 | 60.9 | 69.8 |
| HMAN w/o AE | 25.7 | 46.4 | 57.3 | 25.5 | 64.7 | 57.9 | 29.6 | 55.1 | 66.1 | 29.9 | 56.1 | 67.4 | 32.5 | 59.2 | 67.8 | 32.9 | 59.4 | 68.4 |
| HMAN w/o HW | 25.2 | 46.0 | 57.9 | 25.2 | 45.9 | 57.9 | 28.6 | 52.6 | 62.2 | 28.5 | 53.0 | 63.0 | 32.8 | 60.9 | 70.0 | 32.9 | 60.2 | 70.3 |

Table 4.3: Results of using graph-based information on DBP15K and DBP100K. @1, @10, and @50 refer to Hits@1, Hits@10, and Hits@50, respectively. Each aspect (i.e., topological, relation, and attribute features) and highway layer are individually removed to perform an ablation study, denoted as w/o TE (RE, AE, and HW).

MAN, HMAN shows more significant performance drops, which also demonstrates that employing the feedforward networks can better categorize relation and attribute features than GCNs in this scenario. Interestingly, looking at the two variants MAN w/o TE and HMAN w/o TE, we can see the former achieves better results. Since MAN propagates relation and attribute features via graph structures, it can still implicitly capture topological knowledge of entities even after we remove the topological features. However, HMAN loses such structure knowledge when topological features are excluded, and thus its results are worse. From these experiments, we

| | **English** | **Chinese** |
|---|---|---|
| **ILL pair** | Casino_Royale_(2006_film) (3) | 007大戰皇家賭場 (3) |
| **Features** | starring, starring, distributor | starring, starring, language |
| **Neighbors** | Daniel_Craig (1), Eva_Green (4), Columbia_Pictures (9) | 丹尼爾·克雷格 (1), 伊娃·格蓮 (4), 英語 (832) |

Table 4.4: Case study of the noise introduced by the propagation mechanism.

can conclude that the topological information is playing an indispensable role among the given features.

To understand why HMAN outperforms MAN, in the following we describe a case study to provide insights potentially explaining the performance gap. Recall that MAN collects relation and attribute information by the propagation mechanism in GCNs where such knowledge is exchanged through neighbors, while HMAN uses feedforward networks to capture expressive features directly from the input feature vectors without propagation. As we discussed before, it is not always the case that neighbors of equivalent entities share similar relations or attributes. Propagating such features through linked entities in GCNs may introduce noise and thus harm performance. Table 4.4 presents an example, which is a pair of entities extracted from DBP15K. We use the number in parentheses (*) after entity names to denote the number of relation features they have. In this case, the two entities "*Casino_Royale_(2006_film)*" in the source language (English) and "*007大戰皇家賭場*" in the target language (Chinese) both have three relation features. We notice that the propagation mechanism introduces some neighbors which are unable to find cross-lingual counterparts from the other end, marked in red. Considering the entity "英語" (English), a neighbor of "*007大戰皇家賭場*", no counterparts can be found in the neighbors of "*Casino_Royale_(2006_film)*". We also observe that "英語" (English) is a pivot node in the Chinese KG and has 832 relations, such as "語言" (Language), "官方語言" (Official Language), and "頻道語言" (Channel Language). In this situation, propagating features from neighbors can harm performance. In fact, the feature sets of the ILL pair already convey information that captures their similarity (e.g., the "starring" marked in blue are shared twice). Therefore, by directly using feedforward networks, HMAN is able to effectively capture such knowledge.

## Results with Textual Embeddings

In this subsection, we discuss empirical results involving the addition of entity descriptions, shown in Table 4.5. Applying literal descriptions of entities to conduct cross-lingual entity matching is relatively under-explored. The recent work of Chen et al. [17] used entity descriptions in their model; however, we are unable to make comparisons with their work, as we do not have access to their code and data. Since we employ BERT to learn textual embeddings of descriptions, we consider systems that also use external resources, like Google Translate,[4] as our baselines. We directly take results reported by Sun et al. [92], denoted as "Translation" and "JAPE+Translation".

The POINTWISEBERT model is used with GCN-based models, which largely reduces the search space, as indicated by MAN (RERANK) and HMAN (RERANK), where the difference is that the candidate pools are given by MAN and HMAN, respectively. For DBP15K, we select top-200 candidate target entities as the candidate pool while for DBP100K, top-20 candidates are selected due to its larger size. The reranking method does lead to performance gains across all datasets, where the improvements are dependent on the quality of the candidate pools. HMAN (RERANK) generally performs better than MAN (RERANK) since HMAN recommends more promising candidate pools.

The PAIRWISEBERT model learns the textual embeddings that map cross-lingual descriptions into the same space, which can be directly used to match entities. The results are listed under PAIRWISEBERT in Table 4.5. We can see that it achieves good results on its own, which also shows the efficacy of using multilingual descriptions. Moreover, such textual embeddings can be combined with graph embeddings (learned by MAN or HMAN) by weighted concatenation, as discussed in Section 3.6. The results are reported as MAN (WEIGHTED) and HMAN (WEIGHTED), respectively. As we can see, this simple operation leads to significant improvements and gives excellent results across all datasets.

### 4.3.2 Effect of Unlinkable Cases

We further investigate the efficacy of our models facing the possibility that the cross-lingual counterparts are absent on XEM15. As HMAN and PAIRWISEBERT outperform the others in the tests of DBP15K and DBP100K, we only evaluate these two models in this section. To enable NIL prediction, the most straightforward way is to set a threshold: if the similarity score is lower than such a threshold, the model predicts NIL. Nonetheless, as Rao et al. [78] mentioned in their work, a uniform threshold may not be favorable in practice but determining thresholds for every case is difficult. Therefore, we follow their approach and additionally insert a dummy entity into KGs to represent the absence of counterparts. Given the features of the dummy entity, the ranker can learn to predict NIL as such an "answer" is included in the ranking. For HMAN, we only use the topological features of the dummy entity, i.e., it is treated as a self-connecting node, and no relations or attributes are assigned. With respect to the textual information used for PAIRWISEBERT, the string value "NIL" is provided as its description. The NIL% of each

[4] https://cloud.google.com/translate/

| Model | ZH → EN | | | EN→ ZH | | | JA → EN | | | EN→ JA | | | FR → EN | | | EN→ FR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 | @1 | @10 | @50 |
| | DBP15K | | | | | | | | | | | | | | | | | |
| Translation∗ | 55.7 | 67.6 | 74.3 | 40.3 | 54.2 | 62.2 | 74.6 | 84.5 | 89.1 | 61.9 | 72.0 | 77.2 | - | - | - | - | - | - |
| JAPE + Translation∗ | 73.0 | 90.4 | 96.6 | 62.7 | 85.2 | 94.2 | 82.8 | 94.6 | 98.3 | 75.9 | 90.7 | 96.0 | - | - | - | - | - | - |
| PairwiseBert | 74.3 | 94.6 | 98.8 | 74.8 | 94.7 | 99.0 | 78.6 | 95.8 | 98.5 | 78.3 | 95.4 | 98.4 | 95.2 | 99.2 | 99.6 | 94.9 | 99.2 | 99.7 |
| Man (Rerank) | 84.2 | 93.6 | 94.8 | 82.1 | 91.8 | 93.1 | 89.4 | 94.0 | 94.8 | 88.2 | 93.3 | 94.0 | 93.1 | 95.2 | 95.4 | 93.1 | 95.3 | 95.4 |
| Hman (Rerank) | 86.5 | 95.9 | 96.9 | 85.8 | 94.1 | 95.3 | 89.0 | 96.0 | 97.3 | 89.0 | 96.0 | 97.5 | 95.3 | 97.7 | 97.8 | 95.2 | 97.9 | 98.1 |
| Man (Weighted) | 85.4 | 98.2 | 99.7 | 83.8 | 97.7 | 99.5 | 90.8 | 98.8 | 99.7 | 89.9 | 98.5 | 99.5 | 96.8 | 99.6 | 99.8 | 96.7 | 99.7 | **99.9** |
| Hman (Weighted) | **87.1** | **98.7** | **99.8** | **86.4** | **98.5** | **99.8** | **93.5** | **99.4** | **99.9** | **93.3** | **99.3** | **99.9** | **97.3** | **99.8** | **99.9** | **97.3** | **99.8** | **99.9** |
| | DBP100K | | | | | | | | | | | | | | | | | |
| PairwiseBert | 65.1 | 85.1 | 92.6 | 66.2 | 85.8 | 92.9 | 67.7 | 86.5 | 93.1 | 67.9 | 86.4 | 93.2 | 93.2 | 97.9 | 98.9 | 93.4 | 98.0 | 98.9 |
| Man (Rerank) | 59.5 | 62.1 | 62.2 | 55.9 | 58.2 | 58.2 | 65.5 | 68.2 | 68.4 | 59.9 | 62.1 | 62.3 | 69.7 | 70.4 | 70.5 | 65.5 | 66.2 | 66.2 |
| Hman (Rerank) | 58.9 | 61.2 | 61.3 | 57.9 | 60.2 | 60.3 | 66.9 | 69.4 | 69.6 | 67.0 | 69.6 | 69.8 | 72.1 | 72.9 | 73.0 | 72.7 | 73.5 | 73.5 |
| Man (Weighted) | **81.4** | **94.9** | **98.2** | **80.5** | 94.1 | 97.7 | 84.3 | 95.4 | 98.3 | 81.5 | 94.2 | 97.6 | 96.2 | 99.3 | **99.7** | 95.7 | 99.1 | 99.6 |
| Hman (Weighted) | 81.1 | 94.3 | 97.8 | 80.3 | **94.5** | **97.9** | **85.2** | **96.1** | **98.4** | **84.6** | **96.1** | **98.5** | **96.5** | **99.4** | **99.7** | **96.5** | **99.5** | **99.8** |

Table 4.5: Results of using both graph and textual information on DBP15K and DBP100K. @1, @10, and @50 refer to Hits@1, Hits@10, and Hits@50, respectively. ∗ indicates results are taken from Sun et al. [92].

language is introduced as our Baseline method, which is equivalent to predicting every sample as NIL. The experimental results on XEM15 are displayed in Table 4.6. For simplicity, only *Hits@1* and *Hits@10* are reported.

Three settings of the test are conducted: (1) Linkable: only the entities that have English counterparts are evaluated, which is similar to the traditional settings; (2) Unlinkable: evaluating the entities whose English counterparts are absent, i.e., NIL prediction; (3) All: the combination of (1) and (2). Overall, Hman (Weighted) is the state of the art and makes significant improvement on top of PairwiseBert, which is consistent with the previous conclusion — the proposed strategy is able to efficiently integrate multi-aspect information. Nonetheless, as shown in the results of (3), it is not always the case that Hman beats Baseline, and the main reason is the deficient performance on the unlinkable ones. Moreover, when we look closer at the results of (2), in some cases the performance of PairwiseBert instead drops after concatenating with Hman, e.g., BN and HI. These results reveal that our graph-based method, i.e., the Hman, falls short in identifying NIL cases and a more useful modification other than adding a dummy entity is needed, which we leave for future work.

Comparing by language, both Hman and PairwiseBert perform poorly on KN, which can be explained by the lack of information in the Kannada KG (i.e., the least number of entities and high NIL%). We also perform correlation analysis and it shows a moderate negative (strong

| Model | BN | EL | FA | HE | HI | HY | KA | KN | KO | MK | ML | TA | TL | UR | YUE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Hits@1 | | | | | | | | | | | | | | |
| **Linkable** | | | | | | | | | | | | | | | |
| PairwiseBert | 51.3 | 52.5 | 53.2 | 54.7 | 33.7 | 58.5 | 54.0 | 15.1 | 48.4 | 68.1 | 40.6 | 30.4 | 87.9 | 58.1 | 44.5 |
| Hman | 32.4 | 33.2 | 34.3 | 40.1 | 19.4 | 24.6 | 32.1 | 17.1 | 37.7 | 31.5 | 26.8 | 19.4 | 14.5 | 27.8 | 20.8 |
| Hman (Weighted) | **70.5** | **70.3** | **71.1** | **73.1** | **53.8** | **72.9** | **74.6** | **30.7** | **68.2** | **80.3** | **60.9** | **52.1** | **88.9** | **76.9** | **57.4** |
| **Unlinkable** | | | | | | | | | | | | | | | |
| PairwiseBert | **47.1** | 73.1 | 42.1 | 44.8 | **84.0** | 56.8 | 64.4 | 72.6 | 61.6 | 60.7 | 49.5 | **83.3** | 33.1 | 55.2 | 42.1 |
| Hman | 7.0 | 16.1 | 20.0 | 10.6 | 18.9 | 18.6 | 8.8 | 0.8 | 26.7 | 10.4 | 2.5 | 17.1 | 28.2 | 18.4 | 6.6 |
| Hman (Weighted) | 46.9 | **80.9** | **60.1** | **60.0** | 71.4 | **70.6** | **75.8** | **75.5** | **73.8** | **75.6** | **54.3** | 68.4 | **62.0** | **59.9** | **51.9** |
| **All** | | | | | | | | | | | | | | | |
| Baseline | 17.0 | 25.5 | 14.0 | 21.7 | 44.9 | 38.4 | 26.5 | 35.3 | 35.2 | 30.3 | 22.4 | 37.7 | 12.9 | 16.2 | 25.5 |
| PairwiseBert | 50.6 | 57.8 | 51.6 | 52.6 | 56.2 | 57.9 | 56.7 | 35.4 | 53.0 | 65.9 | 42.6 | 50.4 | 80.9 | 57.7 | 43.9 |
| Hman | 28.1 | 28.8 | 32.2 | 33.8 | 19.2 | 22.3 | 26.0 | 11.3 | 33.8 | 25.0 | 21.3 | 18.6 | 16.2 | 26.2 | 17.2 |
| Hman (Weighted) | **66.5** | **73.0** | **69.5** | **70.3** | **61.7** | **72.0** | **75.0** | **46.5** | **70.2** | **78.9** | **59.4** | **58.3** | **85.5** | **74.1** | **56.0** |
| | Hits@10 | | | | | | | | | | | | | | |
| **Linkable** | | | | | | | | | | | | | | | |
| PairwiseBert | 73.3 | 68.2 | 74.0 | 72.0 | 44.6 | 70.8 | 71.1 | 29.2 | 66.2 | 81.0 | 57.2 | 43.8 | 91.2 | 78.6 | 64.2 |
| Hman | 57.4 | 58.5 | 57.9 | 63.8 | 39.1 | 49.0 | 59.8 | 37.1 | 61.7 | 58.2 | 49.7 | 39.0 | 37.3 | 53.9 | 42.5 |
| Hman (Weighted) | **84.3** | **80.1** | **85.0** | **83.9** | **69.8** | **81.9** | **82.6** | **43.5** | **79.5** | **88.1** | **70.4** | **69.6** | **91.8** | **89.3** | **75.7** |
| **Unlinkable** | | | | | | | | | | | | | | | |
| PairwiseBert | **80.3** | 89.0 | 74.4 | 76.1 | **97.2** | 87.2 | 89.1 | 94.4 | 85.5 | 86.7 | 81.2 | **97.2** | 78.5 | **79.5** | 74.3 |
| Hman | 15.5 | 46.1 | 38.1 | 30.0 | 21.3 | 39.7 | 24.0 | 1.5 | 43.7 | 37.3 | 5.8 | 21.3 | 44.4 | 29.8 | 21.5 |
| Hman (Weighted) | 78.7 | **93.4** | **80.2** | **84.5** | 86.6 | **92.2** | **95.7** | **97.2** | **92.3** | **93.4** | **87.9** | 82.0 | **84.4** | 77.3 | **83.5** |
| **All** | | | | | | | | | | | | | | | |
| Baseline | 17.0 | 25.5 | 14.0 | 21.7 | 44.9 | 38.4 | 26.5 | 35.3 | 35.2 | 30.3 | 22.4 | 37.7 | 12.9 | 16.2 | 25.5 |
| PairwiseBert | 74.5 | 73.5 | 74.1 | 72.9 | 68.2 | 77.1 | 75.8 | 52.2 | 73.0 | 82.7 | 62.6 | 64.0 | 89.6 | 78.8 | 66.8 |
| Hman | 50.3 | 55.3 | 55.1 | 56.6 | 31.2 | 45.4 | 50.4 | 24.5 | 55.4 | 51.8 | 39.9 | 32.3 | 38.2 | 50.0 | 37.2 |
| Hman (Weighted) | **83.4** | **83.5** | **84.3** | **84.0** | **77.3** | **85.8** | **86.0** | **62.5** | **84.0** | **89.7** | **74.3** | **74.3** | **90.8** | **87.4** | **77.7** |

Table 4.6: Results of using NIL labels and the information of low-resource languages on XEM15. Hits@1 and Hits@10 are reported.

positive) relationship between NIL% and the performance of PairwiseBert under the setting of linkable (unlinkable), whose Pearson's r is -0.56 (0.81). This result is as expected, since the higher NIL%, the more difficult (easier) to predict the linkable (unlinkable) cases from the available amount of information. However, a similar pattern is not found on Hman, which further points out that it is unable to efficiently take advantage of more NIL labels. Rather, the performance of Hman highly correlates with the number of entities in a language and the Pearson's r achieves 0.81 while all the cases are tested, i.e., the setting (3). Such a relationship suggests that Hman is sensitive to the sizes of KGs, whereas it is not the case for PairwiseBert.

# Chapter 5

# Conclusion

In this thesis, we focus on the task of cross-lingual entity matching, which aims to discover the mappings of equivalent entities in multilingual knowledge graphs. We proposed two GCN-based models and two uses of multilingual BERT to investigate how to better utilize multi-aspect information of entities provided by KGs, including topological connections, relations, attributes, and entity descriptions. Empirical results demonstrate that our best model consistently achieves state-of-the-art performance on the benchmark datasets. In addition, we create XEM15, a new dataset emphasizing NIL prediction and the comparison across low-resource languages, to bring up the real-world challenges with regard to this research direction.

In future work, we would explore alternative techniques that can leverage NIL labels more efficiently, especially for graph-based methods. One direction is following Rao et al. [78] to design features that encode global information for the dummy entity. However, their proposed features are tailored for SVM, which are not directly compatible with our GCN-based model. To apply a similar idea to graph embedding learning, we may introduce global attention [58] and allow HMAN to learn the features from data. Another naive approach that enables propagating global information to the dummy entity is to build auxiliary relations connecting to every entity in the graph. Apart from the current benchmarks, we also consider extending our models to aligning heterogeneous multilingual KGs, e.g., Wikidata and YAGO, in which the disparity in schemas would bring new challenges. One foreseeable issue is that a supplementary component for relation alignment might be needed. Moreover, to better ensure the quality of XEM15, we would seek more resources, e.g., human translators speaking languages other than Cantonese, for a more thorough verification of the NIL labels.

# References

[1] Dean Allemang and James Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL.* Elsevier, 2011.

[2] Bo An, Bo Chen, Xianpei Han, and Le Sun. Accurate text-enhanced knowledge graph representation learning. In *Proceedings of NAACL*, 2018.

[3] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.

[4] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *arXiv preprint arXiv:1812.10464*, 2018.

[5] Michael Azmy, Peng Shi, Jimmy Lin, and Ihab F. Ilyas. Matching entities across different knowledge graphs with graph embeddings. *arXiv preprint arXiv:1903.06607*, 2019.

[6] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.

[7] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.

[8] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia: A crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[9] 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*, 2008.

[10] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2013.

[11] 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.

[12] Léon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nimes*, 1991.

[13] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[14] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[15] Ramon Ferrer I Cancho and Richard V Solé. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265, 2001.

[16] Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Gaia Varese. Ontology and instance matching. In *Knowledge-driven multimedia information extraction and ontology evolution*. Springer, 2011.

[17] 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. In *Proceedings of IJCAI*, 2018.

[18] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[19] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of IJCAI*, 2017.

[20] Muhao Chen, Tao Zhou, Pei Zhou, and Carlo Zaniolo. Multi-graph affinity embeddings for multilingual knowledge graphs. In *Proceedings of NIPS Workshop on Automated Knowledge Base Construction*, 2017.

[21] Guillem Collell, Ted Zhang, and Marie-Francine Moens. Imagined visual representations as multimodal embeddings. In *Proceedings of AAAI*, 2017.

[22] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP*, 2018.

[23] Kareem Darwish. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of ACL*, 2013.

[24] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of AAAI*, 2018.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, 2019.

[26] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Deeper–deep entity resolution. *arXiv preprint arXiv:1710.00597*, 2017.

[27] Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. Zero-shot cross-lingual classification using multilingual neural machine translation. *arXiv preprint arXiv:1809.04686*, 2018.

[28] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, 9(1):77–129, 2018.

[29] Michael Färber and Achim Rettinger. Which knowledge graph is best for me? *arXiv preprint arXiv:1809.11099*, 2018.

[30] Christiane Fellbaum. Wordnet. *The encyclopedia of applied linguistics*, 2012.

[31] Linton C Freeman. Visualizing social networks. *Journal of social structure*, 1(1):4, 2000.

[32] Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, Yantao Jia, Huawei Shen, Zixuan Li, and Xueqi Cheng. Self-learning and embedding based entity alignment. *Knowledge and Information Systems*, 2019.

[33] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of ACL*, 2019.

[34] Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association of Computational Linguistics*, 7:297–312, 2019.

[35] Scott A Hale. Multilinguals and wikipedia editing. In *Proceedings of the 2014 ACM conference on Web science*, 2014.

[36] Xu Han, Zhiyuan Liu, and Maosong Sun. Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125*, 2016.

[37] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. A joint embedding method for entity alignment of knowledge bases. In *Proceedings of China Conference on Knowledge Graph and Semantic Computing*, 2016.

[38] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[39] He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of ACL*, 2017.

[40] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributed semantics. In *Proceedings of ACL*, 2014.

[41] Marvin Hofer, Sebastian Hellmann, Milan Dojchinovski, and Johannes Frey. The new dbpedia release cycle: Increasing agility and efficiency in knowledge extraction workflows. In *International Conference on Semantic Systems*, pages 1–18. Springer, 2020.

[42] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of IJCNLP*, 2015.

[43] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5:339–351, 2017.

[44] Kaliyaperumal Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*, 2019.

[45] Douwe Kiela and Léon Bottou. Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of EMNLP*, 2014.

[46] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*, 2017.

[47] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. In *Proceedings of NAACL*, 2019.

[48] Pradap Konda, Sanjib Das, Paul Suganthan GC, AnHai Doan, Adel Ardalan, Jeffrey R Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, et al. Magellan: Toward building entity matching management systems. *Proceedings of the VLDB Endowment*, 2016.

[49] Dimitris Kontokostas, Charalampos Bratsas, Sören Auer, Sebastian Hellmann, Ioannis Antoniou, and George Metakides. Internationalization of linked data: The case of the greek dbpedia edition. *Journal of Web Semantics*, 15:51–61, 2012.

[50] 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.

[51] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[52] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*, 2015.

[53] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*, 2015.

[54] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th international conference on World Wide Web*, 2017.

[55] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013.

[56] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of NAACL*, 2018.

[57] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[58] Hesham Mostafa and Marcel Nassar. Permutohedral-gcn: Graph convolutional networks with global attention. *arXiv preprint arXiv:2003.00635*, 2020.

[59] Diego Moussallem, Matthias Wauer, and Axel-Cyrille Ngonga Ngomo. Machine translation using semantic web technologies: A survey. *Journal of Web Semantics*, 51:1–19, 2018.

[60] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of SIGMOD*, 2018.

[61] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[62] Axel-Cyrille Ngonga Ngomo and Soren Auer. Limes: a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.

[63] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of ENMLP*, 2018.

[64] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of NAACL*, 2019.

[65] Dat Quoc Nguyen. An overview of embedding models of entities and relationships for knowledge base completion. *arXiv preprint arXiv:1703.08098*, 2017.

[66] Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of NAACL*, 2016.

[67] 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, 2016.

[68] Feiping Nie, Wei Zhu, and Xuelong Li. Unsupervised large graph embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2422–2428, 2017.

[69] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

[70] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th international conference on world wide web*, pages 1419–1428, 2016.

[71] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[72] Maria Pershina, Mohamed Yakout, and Kaushik Chakrabarti. Holistic entity matching across knowledge graphs. In *Proceedings of IEEE International Conference on Big Data*, 2015.

[73] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL*, 2018.

[74] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.

[75] Alessandro Piscopo. Wikidata: A new paradigm of human-bot collaboration? *arXiv preprint arXiv:1810.00931*, 2018.

[76] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training (2018). *URL* `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`, 2018.

[77] Yves Raimond, Christopher Sutton, and Mark B. Sandler. Automatic interlinking of music datasets on the semantic web. In *Proceedings of WWW workshop on Linked Data on the Web*, 2008.

[78] Delip Rao, Paul McNamee, and Mark Dredze. Entity linking: Finding extracted entities in a knowledge base. In *Multi-source, multilingual information extraction and summarization*, pages 93–115. Springer, 2013.

[79] Jinfeng Rao, Hua He, and Jimmy Lin. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of CIKM*, 2016.

[80] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from Wikipedia, WordNet, and GeoNames. In *Proceedings of International Semantic Web Conference*, 2016.

[81] Petar Ristoski. *Exploiting semantic web knowledge graphs in data mining*, volume 38. IOS Press, 2019.

[82] Amrita Saha, Vardaan Pahuja, Mitesh M Khapra, Karthik Sankaranarayanan, and Sarath Chandar. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[83] François Scharffe, Yanbin Liu, and Chuguang Zhou. RDF-AI: an architecture for RDF datasets matching, fusion and interlink. In *Proceedings of IJCAI workshop on Identity and Reference in Web-based Knowledge Representation*, 2009.

[84] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Proceedings of European Semantic Web Conference*, 2018.

[85] Peng Shi, Jinfeng Rao, and Jimmy Lin. Simple attention-based representation learning for ranking short social media posts. In *Proceedings of NAACL*, 2019.

[86] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[87] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for amr-to-text generation. In *Proceedings of ACL*, 2018.

[88] Dennis Spohr, Laura Hollink, and Philipp Cimiano. A machine learning approach to multilingual and cross-lingual ontology matching. In *Proceedings of International Semantic Web Conference*, 2011.

[89] Rupesh K. Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Proceedings of NIPS*, 2015.

[90] Thomas Steiner. Bots vs. wikipedians, anons vs. logged-ins (redux) a global study of edit activity on wikipedia and wikidata. In *Proceedings of The International Symposium on Open Collaboration*, 2014.

[91] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008.

[92] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *Proceedings of International Semantic Web Conference*, 2017.

[93] Athanasios Theocharidis, Stjin Van Dongen, Anton J Enright, and Tom C Freeman. Network visualization and analysis of gene expression data using biolayout express 3d. *Nature protocols*, 4(10):1535, 2009.

[94] Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of ACL*, 2016.

[95] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2016.

[96] Bayu Distiawan Trsedya, Jianzhong Qi, and Rui Zhang. Entity alignment between knowledge graphs using attribute embeddings. In *Proceedings of AAAI*, 2019.

[97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[98] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of International Semantic Web Conference*, 2009.

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

[100] Andra Waagmeester, Egon L Willighagen, Andrew I Su, Martina Kutmon, Jose Emilio Labra Gayo, Daniel Fernández-Álvarez, Quentin Groom, Peter J Schaap, Lisa M Verhagen, and Jasper J Koehorst. A protocol for adding knowledge to wikidata, a case report. *BioRxiv*, 2020.

[101] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[102] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *AAAI*, volume 17, pages 203–209, 2017.

[103] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*, 2014.

[104] Zhichun Wang, Juanzi Li, Zhigang Wang, and Jie Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of WWW*, 2012.

[105] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of EMNLP*, 2018.

[106] Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of EMNLP*, 2019.

[107] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of AAAI*, 2016.

[108] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*, 2015.

[109] Hsiu-Wei Yang, Yanyan Zou, Peng Shi, Wei Lu, Jimmy Lin, and Xu Sun. Aligning cross-lingual entities with multi-aspect information. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[110] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*, 2018.

[111] Weiguo Zheng, Lei Zou, Wei Peng, Xifeng Yan, Shaoxu Song, and Dongyan Zhao. Semantic sparql similarity search over rdf knowledge graphs. *Proceedings of the VLDB Endowment*, 9(11):840–851, 2016.

[112] Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of EMNLP*, 2015.

[113] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Iterative entity alignment via joint knowledge embeddings. In *Proceedings of IJCAI*, 2017.