UNIVERSITY OF CALIFORNIA

Los Angeles

Multi-relational Representation Learning and Knowledge Acquisition

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Muhao Chen

2019

Abstract of the Dissertation

Multi-relational Representation Learning and Knowledge Acquisition

by

Muhao Chen
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2019
Professor Carlo Zaniolo, Chair

Multi-relational representation learning methods encode entities or concepts of a knowledge graph in a continuous and low-dimensional vector space, where the relational inferences of entities (concepts) are modeled as some simple vector algebras. Despite such knowledge representations being crucial to a wide range of knowledge-driven applications, state-of-the-art methods are limited to learning embeddings for simple relation facts in a single knowledge graph. In this dissertation, we pursue the goal of comprehensively capturing the multifaceted relational knowledge in various types of knowledge bases, and towards that we contribute on three fronts: (i) we introduce the first multi-relational representation learning framework that learns to transfer embeddings across multiple knowledge bases; (ii) we propose techniques for preserving relational facts with complex properties in the embedding space, including those enforce relational properties, form hierarchies, or endowed uncertainty; (iii) we investigate large-scale relational learning based on other modalities of data, with the aim of acquiring knowledge to enrich the knowledge bases.

Each of these three research problems presents a series of key challenges which we address. Thus, for transferred embeddings, we develop joint learning of relational structure encoders that confront the heterogeneity of contents in knowledge graphs, together with diverse types of alignment models that learn to transfer on the basis of simple, hierarchical or fuzzy alignment information. In addition, we extend the joint learning framework with semi-supervised co-training of entity descriptions, and proactive score propagation for fuzzy alignment, so as to conquer the scenarios where alignment information is limitedly provided. To capture complex relation facts, we focus first on the relational properties that cause non-linearity in embedding structures, for which we leverage a non-linear component-specific mappings of embeddings to eliminate the conflicts, and strengthens the learning process with hierarchical regularization. For uncertain relation facts, we preserve the uncertainty by utilizing Probablistic Soft Logic to guide the non-linear regressor that is jointly trained with the structure encoder. We further study the support of relational learning based on sequence data. Our model proposes generic neural sequence pair models to support large-scale relation detection, in which we incorporate different sequence encoders for heterogeneous data such as structured articles, amino acid sequences, and lexicographic knowledge.

The methods proposed in this dissertation extend the application of multi-relational embeddings, and improve a wide spectrum of applications in different domains. These include knowledge alignment, monolingual and cross-lingual knowledge graph completion, semantic search, entity typing, paraphrase identification, uncertain relation prediction, protein-protein interaction prediction, protein binding affinity estimation, single-cell RNA-sequence imputation, and Web-scale sub-article matching.

The dissertation of Muhao Chen is approved.

Junghoo Cho

Yizhou Sun

Yingnian Wu

Carlo Zaniolo, Committee Chair

University of California, Los Angeles

2019

To life.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Challenges 3

1.3 Thesis Contributions 5

1.4 Thesis Overview 7

2 Background **11**

2.1 Multi-relational Representation Learning 11

2.2 Knowledge Bases and Knowledge Graphs 13

2.2.1 Monolingual and Multilingual Knowledge 13

2.2.2 Ontology and Instance-level Knowledge Graphs 13

2.2.3 Comprehensive Properties of Relation Facts 14

2.2.4 Sequence Data As Side Information 16

2.3 Neural Sequence Models 17

2.3.1 The Convolution Layer with Pooling 17

2.3.2 GRU and Residual GRU 17

2.3.3 Self-attentive Encoder 19

2.3.4 Linear Bag-of-words 20

3 Transfer Embeddings of Multilingual Knowledge Graphs 21

3.1 Introduction 21

3.2 Related Work 24

3.3 The Vanilla Multilingual Knowledge Graph Embeddings 25

3.3.1 Multilingual Knowledge Graphs 25

3.3.2 Knowledge Model 26

3.3.3 Alignment Model 26

3.3.4 Variants of `MTransE` 28

3.4 Cross-lingual Entity Matching 30

3.4.1 Examples of Knowledge Alignment 33

3.5 Semi-supervised Co-training 35

3.5.1 Iterative Co-training 37

3.6 Semi-supervised and Zero-shot Entity Alignment 38

3.6.1 Cross-lingual Entity Alignment 39

3.6.2 Zero-shot Alignment 41

3.6.3 Cross-lingual KG Completion 42

3.7 Conclusion 43

4 Transfer Embeddings with Complex Alignment Information 44

4.1 Introduction 44

4.1.1 Ontology-level Concepts and Instance-level Entities 44

4.1.2 Single-cell RNA-sequence Data As Fuzzy Alignment 47

4.2 Joint Embedding of Ontological Concepts and Instance-view Entities 48

4.2.1 Formalization of Two-view Knowledge Bases 48

4.2.2 Alignment Model with Hierarchical Grouping Techniques 49

4.2.3 Knowledge Model . . . 52

4.2.4 Joint Training on Two Views . . . 55

4.2.5 Variants of JOIE and Complexity . . . 55

4.3 Entity Typing and Triple Completion . . . 56

4.3.1 Datasets . . . 56

4.3.2 Triple Completion . . . 57

4.3.3 Entity Typing . . . 59

4.3.4 Case Study . . . 61

4.4 Ablation Study . . . 64

4.4.1 Dimensionality . . . 65

4.4.2 Sufficiency of Type Information . . . 66

4.4.3 Effects of Negative Sampling . . . 67

4.5 Transfer Gene Knowledge Based on Fuzzy Alignment . . . 68

4.6 Cell Clustering . . . 70

4.6.1 Dataset . . . 70

4.6.2 Evaluation . . . 71

4.7 Conclusion . . . 72

5 Learning to Capture Relational Properties . . . 74

5.1 Introduction . . . 74

5.2 Embedding Ontology Graphs With Complex Relational Properties . . . 76

5.2.1 Preliminary . . . 76

5.2.2 Modeling . . . 76

5.2.3 Learning Process . . . 81

5.3 Experiments . . . 82

5.3.1 Relation Prediction 82

5.3.2 Relation Verification 85

5.3.3 Case Study 87

5.4 Conclusion 88

6 Embedding Uncertain Knowledge Graphs 89

6.1 Introduction 89

6.2 Related Work 91

6.3 Problem Definition 92

6.4 Modeling 92

6.4.1 Modeling Plausibility for Relation Facts 93

6.4.2 Inferring Confidence Scores for Unseen Relation Facts 94

6.4.3 Embedding Uncertain Knowledge Graphs 97

6.4.4 Model variants 98

6.4.5 Confidence score prediction 100

6.5 Experiments 100

6.5.1 Datasets 100

6.5.2 Model Configurations 103

6.5.3 Confidence Prediction 104

6.5.4 Relation Fact Ranking 105

6.5.5 Relation Fact Classification 107

6.6 Conclusion and Future Work 108

7 Large-scale Sub-article Relation Learning 109

7.1 Introduction 109

7.2 Related Work 112

7.3 Modeling 113

7.3.1 Preliminaries 113

7.3.2 Document Encoders 114

7.3.3 Explicit Features 115

7.3.4 Training 116

7.4 Experiments 117

7.4.1 Evaluation 119

7.4.2 Mining Sub-articles from the Entire English Wikipedia 123

7.5 Conclusion and Future Work 124

8 Multifaceted Protein-Protein Interaction Learning 125

8.1 Introduction 125

8.2 Related Work 127

8.3 Methods 129

8.3.1 Preliminary 129

8.3.2 RCNN-based Protein Sequence Encoder 130

8.3.3 Learning Architecture and Learning Objectives 132

8.4 Experiments 133

8.4.1 Datasets 134

8.4.2 Binary PPI Prediction 135

8.4.3 Interaction Type Prediction 139

8.4.4 Binding Affinity Estimation 141

8.4.5 Amino Acid Embeddings 142

8.4.6 Hyperparameter Study 143

8.4.7 Run-time Analysis 144

8.5 Conclusion 144

9 Learning Multi-granular Associations of Lexemes and Sentences 146

9.1 Introduction 146

9.2 Related Work 148

9.3 Modeling Bilingual Dictionaries 150

9.3.1 Encoders for Bilingual Dictionaries 151

9.3.2 Basic Learning Objective 151

9.3.3 Bilingual Multi-task Learning 152

9.3.4 Joint Learning Objective 153

9.4 Experiments 153

9.4.1 Cross-lingual Reverse Dictionary Retrieval 155

9.4.2 Bilingual Paraphrase Identification 157

9.5 Conclusion 159

10 Conclusion and Future Work 160

10.1 Summary 160

10.2 Future Directions 162

List of Figures

2.1 An example of two-view KB. Regular meta-relations and those conforming the hierarchical property are denoted as black and orange dashed lines respectively in the ontology view. 14

3.1 A simple example which shows triples, an ILL, and entity descriptions in a multilingual KG (DBpedia). The French description for *astronome* means *an astronomer is a scientist specialized in the study of astronomy*, which contains much fewer content details than the English description for *astronomer*. 23

3.2 Precision-recall curves for cross-lingual entity matching on WK3l-15k. 31

3.3 Visualization of the result of Var_4 for the same six English entities in their original space (left) and in French space after being transformed (right). English entities are rendered in blue, and the corresponding French entities are in light ruby. 35

4.1 The scRNA-seq fuzzy alignment between genes and cells. 46

4.2 `JOIE` learns two aspects of a knowledge base. The alignment model learns embeddings from type links (dash arrows in green "category" box). The default knowledge model learns embeddings from triples (grey box) in each view; Besides, hierarchy-aware knowledge models the meta-relation facts that form hierarchies in the ontology (orange "Hierarchy" trapezoid). 49

4.3 Intuition of the alignment model: Cross-view Grouping (a); Cross-view Transformation (b). 50

4.4 Examples of ontology population by finding the closest relations in the instance view for the query "Office Holder-Country". Top 10 predicted relations are plotted with their ranks. 63

4.5 Performance of entity typing task on both datasets with different entity and concept embedding dimensionalities 67

4.6 The effect of training the model using different proportions of type links on (a) YAGO26K-906 and (b) DB111K-174 68

4.7 Visualizing the effects on embeddings of negative sampling on type links 69

4.8 ARI of cell clustering under different drop-out rates of the Zeisel dataset. 71

5.1 Depiction of the conflicts of the relation-specific projection for learning transitive relations (Case 1, left), and symmetric relations (Case 2, right). 77

5.2 Precision-recall curves for relation prediction on YG15k and YG60k. 83

5.3 Choices of α_1 values and corresponding accuracy of relation prediction on YG15k. 86

6.1 Precision-Recall curves for relation fact classification task. 107

7.1 The main-article *Bundeswehr* (armed forces of Germany) and its sub-articles. . . . 110

7.2 Learning architecture of the model. 117

7.3 Relative importance (RI) of features analyzed by Garson's algorithm. RI of each feature is aggregated from all folds of cross-validation. 121

7.4 Sensitivity of CNN+F on the proportion of dataset for evaluation. 121

8.1 The overall learning architecture of our framework. 131

8.2 The structure of our residual RCNN encoder is shown on the right, and the RCNN unit is shown on the left. Each RCNN unit contains a convolution-pooling layer followed a bidirectional residual GRU. 131

8.3 Mutation effects on structure and binding affinity. The blue entity is Subtilisin BPN’ precursor (Chain E), and the red entity is Chymotrypsin inhibitor (Chain I). The mutation is highlighted in yellow. The wild type (1TM1) and mutant (1TO1) complexes are retrieved from PDB. 142

8.4 Performance evaluation on dimensionality of hidden states, and the number of occurrences of the RCNN units. 144

9.1 An example illustrating the two cross-lingual tasks. The *cross-lingual reverse dictionary retrieval* finds cross-lingual target words based on descriptions. In terms of *cross-lingual paraphrases*, the French sentence (which means *any male being considered in relation to his father and mother, or only one of them*) describes the same meaning as the English sentence, but has much more content details. 148

9.2 Joint learning architecture of `BilDRL`. 150

List of Tables

2.1 The number of triples of each relation type in Yago3 Ontology. 15

3.1 Summary of model variants. 29

3.2 Statistics of the WK3l datasets. 30

3.3 Number of entity inter-lingual links (ILLs). 30

3.4 Cross-lingual entity matching results. 31

3.5 Examples of cross-lingual entity matching. 33

3.6 Examples of cross-lingual relation matching. 33

3.7 Examples of cross-lingual triple completion. 34

3.8 Statistics of the Wk3l60k dataset. 40

3.9 Results of cross-lingual entity alignment. 40

3.10 Results of zero-shot alignment. 41

3.11 Results of KG completion. 41

4.1 Statistics of datasets. 57

4.2 Results of triple completion. Note that H@1 and H@10 denote $Hit@1$ and $Hit@10$ respectively. For each group of model variants with the same knowledge encoding techniques, the best results are bold-faced. The overall best results on each dataset are under-scored. 59

4.3 Results of entity typing. 60

4.4 Examples of ontology population from JOIE-TransE-CT. Top 5 Populated Triples with smallest L2-norm distances are provided with reasonable answers bold-faced. 64

4.5 Results of long-tail entities typing. 64

4.6 Examples of long-tail entity typing. Top 3 predictions are provided with the correct type bold-faced. 65

4.7 Effects of negative sampling in type links 66

4.8 ARI of cell clustering by KG-Transfer verses the best of all baselines under different drop-out rate. 72

5.1 Model complexity: number of parameters for optimization, and the computational complexity for predicting a relation. n_c and n_r are numbers of concepts and relations, and k is the dimensionality of embeddings. 80

5.2 Statistics of the data sets. pct. prop. and pct. hier. are the percentages of triples defined with relational properties and hierachies. 80

5.3 Accuracy of Relation Prediction (%). prop. means with properties, hier. means hierarchical relations. 83

5.4 Accuracy of relation verification (%). prop. means with properties, hier. means hierarchical relations. 85

5.5 Examples of top-ranked new relation facts. The italic ones are conceptually close. The rest are correct. 86

6.1 Statistics of the extracted datasets used in this chapter. Avg(s) and Std(s) thereof are the average and standard deviation of confidence scores. 101

6.2 Examples of logical rules 101

6.3 Mean square error (MSE) and mean absolute error (MAE) of relation fact confidence prediction. 101

6.4 Examples of confidence prediction using $\mathtt{UKGE}_2$ 102

6.5 Mean Spearman's Rho between predicted rankings and true rankings for relative ranking task. 102

6.6 Mean normalized DCG for global ranking task. Here *linear* stands for linear gain, and *exp.* stands for exponential gain. 102

6.7 Examples of relation fact ranking (global) results using $\texttt{UKGE}_3$ 103

6.8 F-1 scores and accuracies (%) of relation fact classification 103

7.1 Statistics of the dataset. 120

7.2 Cross-validation results on WAP196k. We report precision, recall and F1-scores on three groups of models: (1) statistical classification algorithms based on explicit features, including logistic regression, Naive Bayes Classifier (NBC), Linear SVM, Adaboost (SAMME.R algorithm), Decision Tree (DT), Random Forest (RF) and k-nearest-neighbor classifier (kNN); (2) three types of document pair encoders without explicit features; (3) the proposed model in this chapter that combines explicit features with convolutional document pair encoders (CNN+F), GRU encoders (GRU+F) or attentive encoders (AGRU+F). 120

7.3 Ablation on feature categories for CNN+F. 121

7.4 Examples of recognized main and sub-article matches. The italicize sub-article titles are without overlapping tokens with the main article titles. 123

8.1 Evaluation of binary PPI prediction on the Yeast dataset based on 5-fold cross-validation. We report the mean and standard deviation for the test sets. 136

8.2 Statistical assessment (t-test; two-tailed) on the accuracy of binary PPI prediction. The statistically significant differences are highlighted in red. 136

8.3 Evaluation of binary PPI prediction on variants of multi-species (C. elegan, Drosophila, and E. coli) dataset. 138

8.4 Accuracy (%) and fold changes over zero rule for PPI interaction type prediction on two STRING datasets based on 10-fold cross-validation. 139

8.5 Results for binding affinity prediction on the SKEMPI dataset. Each measurement is an average of the test sets over 10-fold cross-validation. 139

8.6 Comparison of amino acid representations based on binary prediction. 143

8.7 Run-Time of training embeddings and different prediction tasks. 143

9.1 Statistics of the bilingual dictionary dataset Wikt3l. 154

9.2 Examples of bilingual paraphrases from WBP3l. 155

9.3 Cross-lingual reverse dictionary retrieval results by `BilDRL` variants. We report $P@1$, $P@10$, and MRR on four groups of models: (i) basic dictionary models that adopt four different encoding techniques (BOW, CNN, GRU and ATT); (ii) models with the two best encoding techniques that enforce the monolingual retrieval approach by Hill et al. (Hill et al., 2016) (GRU-mono and ATT-mono); (iii) models adopting bilingual multi-task learning (GRU-MTL and ATT-MTL); (iv) joint learning that employs the best dictionary model of ATT-MTL (ATT-joint). 156

9.4 Accuracy and F1-scores of bilingual paraphrase identification. For `BilDRL`, the results by three model variants are reported: `BilDRL`-GRU-MTL and `BilDRL`-ATT-MTL are models with bilingual multi-task learning, and `BilDRL`-ATT-joint is the best ATT-based dictionary model variant deployed with both multi-task and joint learning. 158

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. Carlo Zaniolo, who has guided me throughout the entire Ph.D. study, and inspired me to be an enthusiastic scholar. He has given me enough freedom to explore problems according to my own interests. He has demonstrated all the skills to be a remarkable scholar and a passionate educator.

I would like to thank my committee, Prof. Junghoo Cho, Prof. Yingnian Wu, and Prof. Yizhou Sun, for their advice and suggestions over the years. I am thankful to Prof. Kai-Wei Chang, Prof. Wei Wang, and Prof. Mario Gerla, for closely collaborating with me on many exciting research projects.

I would like to thank all my friends and fellow students at UCLA ScAi Lab, UCLA NLP Group and UCLA NRL Group for their friendship and collaboration. I am sure I am forgetting some who helped me along the way, but this list includes Wasi Uddin Ahmed, Xuelu Chen, Hsuan Chiu, Manoj Reddy Dareddy, Ariyam Das, Ling Ding, Pengyuan Du, Changjun Fan, Shi Gao, Jiaqi Gu, Junheng Hao, Kuan-Hao Huang, Jyun-Yu Jiang, Chelsea Ju, Youfu Li, Ruirui Li, Mingda Li, Giuseppe M. Mazzeo, Tao Meng, Md Rizwan Parvez, Weijia Shi, Alexander Shkapsky, Ankith Uppunda, Jin Wang, Zheng Wang, Zijun Xue, Mohan Yang, Seunghyun Yoo, Jieyu Zhao, Qi Zhao, Guangyu Zhou, and Pei Zhou.

I am grateful to my parents for their unconditional love. Their sacrifice and support have led to all my accomplishments. I also appreciate Jessie Du for accompanying me through most of the time during my Ph.D. study, for which I will always remember. Lastly, I would like to thank Galneryus and Gyze for their great songs that have been encouraging me all the time.

Vita

2014	Bachelor of Science in Computer Science, Fudan University, Shanghai, China.
2015	Research Intern, Teradata Labs, Los Angeles, CA.
2015-2016	Teaching Assistant, Department of Computer Science, UCLA.
2016-2017	Teaching Associate, Department of Computer Science, UCLA.
2016	SDE Intern, Google, Mountain View, CA.
2017-2018	Teaching Fellow, Department of Computer Science, UCLA.
2017	RSDE Intern, Google, Mountain View, CA.
2018	Research Intern, Microsoft Research, Redmond, WA.
2018-2019	Dissertation Year Fellow, UCLA.

PUBLICATIONS

Parts of the work in this dissertation have appeared in the following publications:

- Muhao Chen, Chelsea J. T. Ju, G. Zhou, T. Zhang, Kai-Wei Chang, Carlo Zaniolo, Wei Wang. Multifaceted Protein-Protein Interaction Prediction Based on Siamese Residual RCNN. *Bioinformatics*, Oxford Academic. 2019. *Special issues featuring ISMB/ECCB 2019*
- Muhao Chen, Yingtao Tian, Haochen Chen, Kai-Wei Chang, Steven Skiena, Carlo Zaniolo. Learning to Represent Bilingual Dictionaries. *Arxiv Preprint*. 2019
- Junheng hao, Muhao Chen, Wenchao Yu, Yizhou Sun, Wei Wang. Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts. *KDD*. 2019
- Muhao Chen, Chris Quirk. Embedding Edge-attributed Relational Hierarchies. *SIGIR*. 2019
- Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, Carlo Zaniolo. Embedding Uncertain Knowledge Graphs. *AAAI*. 2019
- Muhao Chen, Changping Meng, Gang Huang, Carlo Zaniolo. Neural Article Pair Modeling for Wikipedia Sub-article Matching. *ECML-PKDD*. 2018
- Muhao Chen, Yingtao Tian, Kai-Wei Chang, et al. Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-lingual Entity Alignment. *IJCAI*. 2018
- Muhao Chen, Yingtao Tian, Xuelu Chen, Z. Xue, Carlo Zaniolo. On2Vec: Embedding-based Relation Prediction for Ontology Population. *SDM*. 2018
- Muhao Chen, Yingtao Tian, Mohan Yang, Carlo Zaniolo. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. *IJCAI*. 2017
- Muhao Chen, Carlo Zaniolo. Learning Multi-faceted Knowledge Graph Embeddings for Natural Language Processing. *IJCAI*. 2017
- Muhao Chen, Tao Zhou, Pei Zhou, Carlo Zaniolo. Multi-graph Affinity Embeddings for Multilingual Knowledge Graphs. *AKBC*. 2017

CHAPTER 1

Introduction

Knowledge bases form large collections of multi-relational data, which model the relations of entities and concepts as large graph structures (i.e. knowledge graphs). The knowledge graph provides a shared understanding of human knowledge that supports commonsense reasoning (Bollacker et al., 2008; Cilibrasi and Vitanyi, 2007; Lehmann et al., 2015; Mahdisoltani et al., 2015; Speer et al., 2017), and also supports domain-specific research work with valuable expert knowledge (Moal and Fernández-Recio, 2012; Szklarczyk et al., 2016; Yu et al., 2016).

1.1 Motivation

This dissertation focuses on representation learning methods for multi-relational data. Multi-relational representation learning models (a.k.a. knowledge graph embeddings) typically encode entities or concepts of a knowledge graph in a continuous and low-dimensional vector space, where the relational inferences of entities (concepts) are modeled as some simple vector algebras. Hence, these models provide efficient and versatile methods to incorporate the symbolic knowledge of multi-relational data into machine learning. Models of this kind have effectively supported a few tasks like knowledge graph completion (Bordes et al., 2013), relation extraction from text (Wang et al., 2014b), and logic rule mining (Yang et al., 2015d). The embedding representations obtained from these models have been indispensable to support a long list of knowledge-driven applications in different domains, including dialogue agents (He et al., 2017a), question answering (Das et al., 2018; Huang et al., 2019), item recommendation (He et al., 2017b), visual object detection (Fang et al., 2017), story understanding (Chen et al., 2019a), drug discovery (Scott et al., 2016) and drug side effect detection (Zitnik et al., 2018).

Despite multi-relational representation learning having been widely utilized, previous methods merely focus on embedding the data in one knowledge graph. In contrast, learning *transferable* embeddings across multiple knowledge graphs represents an unresolved problem. In fact, in many application scenarios, different knowledge graphs can be naturally connected to each other. For example, while multilingual knowledge bases separately manage knowledge graphs in different languages (Lehmann et al., 2015; Mahdisoltani et al., 2015), all those language-specific knowledge graphs are expected to describe consistent relation facts for a shared set of entities in the same domain. In a commonsense ontology where semantic relations seek to provide more abstract concepts (Speer et al., 2017), where each concept can be instantiated into more specific entities in a larger instance-level knowledge graph (Lehmann et al., 2015). Biological data may also be interchangeable across different domains, for instance, interaction data of proteins (Szklarczyk et al., 2016), gene ontologies (Ashburner et al., 2000), and single-cell descriptions (Bard et al., 2005).

Learning embeddings that transfer between different knowledge graphs undoubtedly provides a more generic way to represent knowledge. One immediate benefit from such transferable embeddings is to match and synchronize the entity contents of different language-specific knowledge graphs. This directly extends the use of knowledge graph embeddings to address knowledge alignment tasks (Vulić and Moens, 2015), and is particularly useful to various cross-lingual NLP tasks, as well as other tasks such as entity typing and data cleaning. Moreover, the supported knowledge transfer can lead to an effective way to populate the missing knowledge in one domain based on the knowledge from others. Therefore, knowledge bases of low-resource languages can be easily populated using the embeddings trained on well-populated high-resource ones, without abundant alignment information. For biology research, cell identification and clustering tasks may also borrow knowledge from protein knowledge graphs and gene ontologies, based on partially complete single-cell RNA sequencing transcripts (Elyanow et al., 2019; Lin et al., 2017a).

Besides transferable representation learning, there are other critical problems that are related to multi-relational data. One is that knowledge graphs often enforce complex properties on relation facts. These properties include relational properties such as transitivity and symmetry, as well as hierarchical relations that may be simultaneously endowed transitivity and multi-mapping. Some knowledge graphs also endow uncertainty to relation facts (Mitchell et al., 2018; Speer et al., 2017; Wu et al., 2012). Failure of preserving these important properties in capturing the relations will no doubt cause an imprecise representation of multi-relational knowledge, and further impairs the performance of downstream tasks that are based on the embedding techniques. However, few efforts have investigated the embedding representations needed for relations with these complex properties.

Aside from these issues, knowledge graphs are often far from complete due to the high cost of acquiring high quality relational knowledge (Culotta and Sorensen, 2004; Fundel et al., 2007; Mousavi et al., 2014b). Hence, another important mission is to investigate how to best support relational learning based on data modalities other than graphs, and leverage the acquired knowledge to populate knowledge graphs themselves. As many real-world objects can be represented with sequence information, we investigate relational learning with sequence data.

In this dissertation, we aim at investigating new multi-relational representation learning methods, which produce transferable embeddings across multiple knowledge graphs in different domains, and capture complex properties of the relation facts. For knowledge acquisition, we study the relational learning approaches based on sequence data.

1.2 Challenges

We address several key challenges in this dissertation. First, we propose the study of learning transferable embeddings for different knowledge graphs. Models should characterize the heterogenous structures of different knowledge graphs in the embedding space, while capturing the precise correspondences of entities and relations across graphs. In addition, this learning process is often subject to insufficient supervision, since the alignment information to learn the correspondence is provided to only a limited extent. Second, capturing complex properties of relation facts is nontrivial. Dedicated learning techniques need to be carefully designed to preserve specific relational properties, as well as exploit and propagate uncertainty of relation facts in the embedding space.

Lastly, as knowledge graph themselves are often far from complete, it is vital to populate the graphs by acquiring knowledge from other modalities (both same or different) of data. The challenges we tackle in this dissertation are summarized below:

- **Learning transferred embeddings.** To realize universal and transferable embedding representations for different knowledge graphs, a model should address the following major difficulties of representation learning in two aspects.

  (i) *Heterogeneity in knowledge graph contents.* Different knowledge graphs often lead to heterogeneity in relation facts, as well as that in vocabularies of entities and relations. In fact, different language-specific versions of knowledge graphs in a knowledge base are typically extracted from independently maintained corpora (Lehmann et al., 2015; Mahdisoltani et al., 2015), which no doubt leads to all the aforementioned heterogeneity. If we consider transferring across different domains of knowledge, such as genes and cells (Elyanow et al., 2019) or instances and ontological concepts (Lehmann et al., 2015), then, the relation facts and objects to align between different graphs can even be disjoint with each other.

  (ii) *Diverse types of alignment information.* Diverse types of alignment information are often provided across different knowledge graphs under different application scenarios as well. These include one-to-one alignment between different language-specific knowledge graphs, many-to-one alignment between instance-level entities and ontological concepts (Lehmann et al., 2015), and fuzzy alignment, such as single-cell RNA sequencing transcripts to associate between genes and cells. Each such type of alignment information requires a dedicated method to represent the knowledge transfer.

- **Transfer with limited alignment information.** The information on seed alignment to connect between different knowledge graphs is often very limited. For example, the cross-lingual alignment between language-specific version of a Wikipedia-based knowledge base typically cover less than 20% of entities in each language (Chen et al., 2018e). The alignment information between gene knowledge graphs and cells can be even sparser due to the high cost of verifying single-cell gene expressions in wet lab experiments (Elyanow et al., 2019; Lin et al., 2017a).

Hence, the transfer learning based on limited supervision non-trivially demands carefully designed semi-supervised learning techniques.

- **Learning to capture complex properties of relation facts.** Given one knowledge graph, previous works typically focus on capturing simple deterministic relation facts, and do not learn to preserve the complex properties that are often endowed to relation facts. Towards more precise relational learning, we discuss the challenges in learning two types of properties.

  (i) *Relational properties*. Many knowledge graphs, typically domain-specific ontology graphs, often add relational properties to the knowledge, such as transitivity and symmetry. Some of such relation facts often form hierarchies. A typical example is provided by *Is-A*, which is both transitive and hierarchical, and is the most frequently appearing semantic relation in ontologies. More examples are discussed in Section 2.2.3.1. Such relational facts lead to non-linearity of the embedding structure that cause conflict in a regular Euclidean embedding space.

  (ii) *Uncertainty*. Probablistic knowledge graphs (Wu et al., 2012), commonsense knowledge graphs (Mitchell et al., 2018; Speer et al., 2017) and protein-protein interaction knowledge graphs (Szklarczyk et al., 2016) endow relation facts with uncertainty. As existing multi-relational representation learning methods merely captures deterministic knowledge, the vector algebras employed in these methods do not preserve uncertainty.

- **Relational learning based on sequence data.** The challenges here lie in several aspects. One is that, to represent objects based on sequence data in different application domains, we may have diverse requirements of feature selection from different forms of sequence data, including structured articles, short sentences, and nucleotide sequences. Another issue is how to support large-scale relation detection based on sequence pairs. In some cases, the relation may also appear as a multi-granular association between sequences and lexical units of a sequence.

1.3 Thesis Contributions

- In this dissertation, we propose the first method to learn transferred embeddings across different knowledge graphs. Our learning framework examines multiple techniques to capture cross-lingual knowledge transfer under the simplest one-to-one alignment setting, and identifies the most suitable technique. Our framework is further extended to support knowledge transfer under many-to-one and fuzzy alignment settings. This is accomplished by respectively incorporating hierarchical-grouping based alignment techniques and Semi-non-negative Matrix Tri-facterization based techniques.

- To address the challenging semi-supervised transfer learning of knowledge graph embeddings, this dissertation proposes an iterative co-training model, which leverages the side information of entity descriptions to bridge across different knowledge graphs. For fuzzy alignment settings, our method addresses this problem with proactive score propagation.

- This dissertation proposes two models to capture the two aspects of complex relation properties. The first proposed model leverages a non-linear, component-specific mapping of entity (concept) embeddings to eliminate the conflicts of the embedding structures caused by relational properties, and strengthens the learning of hierarchical relations with hierarchical regularization. To preserve information about uncertainty, the second model jointly learns a non-linear regressor with a multi-relational structure encoder. We also incorporate Probablistic Soft Logic rules into the learning process, to estimate effectively the uncertainty of unseen relation facts through guided confidence score propagation.

- To support relational learning based on sequence data, our work proposes generic neural sequence pair models to support large-scale relation detection between articles, and multi-faceted interaction prediction between proteins. We also propose a joint learning framework that learns the multi-granular association of lexical and sentential semantics in different languages.

- The methods introduced in this dissertation benefit a wide spectrum of applications in different domains. This long list includes knowledge alignment, monolingual and cross-lingual knowledge graph completion, semantic search of entities, entity typing, paraphrase identification, uncertain relation prediction, protein-protein interaction prediction, protein binding affinity estimation, single-cell RNA-sequence imputation, and sub-article matching.

1.4 Thesis Overview

The dissertation can be broadly categorized into five logical segments.

Part I: Background

- Chapter 2:

  The first part of this dissertation surveys background. We start by introducing different categories of multi-relational data. Then we provide an overview of previously proposed knowledge graph embedding approaches. Our work seeks to extend new approaches to support transferable embeddings in (Part II), and to capture complex relation properties in (Part III). Lastly, we define different types neural sequence encoders, which are the basis of our work on sequence-based relational learning in (Part IV), and also constitute the key component of the co-training framework in Chapter 3.

Part II: Transfer Multi-relational Embeddings

- Chapter 3:

  This chapter introduces the first method to learn transferred embeddings of multi-relational data. The proposed model `MTransE` learns to transfer across different language-specific versions of knowledge graphs. `MTransE` uses a combination of two component models, namely the knowledge model and the alignment model. The knowledge model is responsible for encoding entities and relations in a language-specific version of knowledge graph. We explore the method that organizes each language-specific version in a separated embedding space. On top of that, the alignment model learns cross-lingual transitions for both entities and relations across different embedding spaces. We explore the following three representation techniques for cross-lingual alignment: distance-based axis calibration, translation vectors, and linear transformations. To improve the transfer learning under limited supervision, we extend `MTransE` to a novel co-training-based approach `KDCoE` . `KDCoE` iteratively trains `MTransE` and a bilingual entity description embedding model. Starting with a very small portion of entity alignment, both model components alternately propose a set of most confident new ILLs to strengthen the supervision of cross-lingual learning, which leads to gradually improved accuracy on cross-lingual inferences.

- Chapter 4:

  We extend the techniques in the previous chapter to deal with more complex knowledge transfer. JOIE extends `MTransE` to learn the association between instance-level entities and concepts in a hierarchical ontology, for which two types of hierarchical grouping based alignment models are incorporated to capture the many-to-one associations between entities and concepts. `KG-Transfer` modifies `MTransE`'s alignment model as a Semi-non-negative Matrix Tri-factorization technique, so as to capture and propagate the fuzzy alignment information between genes and cells in single-cell RNA sequencing data.

Part III: Learning to Capture Complex Properties of Multi-relational Data

- Chapter 5:

  We study how to preserve relational properties of an ontology with a multi-relational embedding model. The proposed `On2Vec` model integrates two model components that effectively characterize comprehensive relation facts in ontology graphs. The first is the Component-specific Model that encodes concepts and relations into low-dimensional embedding spaces without a loss of relational properties. The second is the Hierarchy Model that performs focused learning of hierarchical relation facts.

- Chapter 6:

  The proposed uncertain KG embedding model `UKGE` seeks to preserve both structural and uncertainty information of relation facts in the embedding space. Unlike previous models that characterize relation facts with binary classification techniques, `UKGE` learns embeddings according to the confidence scores of uncertain relation facts. To further enhance the precision of `UKGE`, we also introduce Probabilistic Soft Logic to infer confidence scores for unseen relation facts during training.

Part IV: Knowledge Acquisition With Neural Sequence Pair Models

- Chapter 7:

  We provide an approach to large-scale detection of the main and sub-article relations for Wikipedia articles. The proposed model adopts a hierarchical learning structure that combines multiple variants of neural document pair encoders with a comprehensive set of explicit features. A large crowdsourced dataset is created to support the evaluation and feature extraction for the task. Based on this large dataset, the proposed model achieves promising results of cross-validation and significantly outperforms previous approaches. Large-scale serving on the entire English Wikipedia also proves the practicability and scalability of the proposed model by effectively extracting a vast collection of newly paired main- and sub-articles.

- Chapter 8:

  This chapter presents an end-to-end framework, `PIPR`, for protein-protein interaction (PPI) predictions using only the protein sequences. `PIPR` incorporates a deep residual recurrent convolutional neural network in the Siamese architecture, which leverages both robust local features and contextualized information, which are significant for capturing the mutual influence of proteins sequences. `PIPR` eliminates the data pre-processing efforts that are required by other systems, and generalizes well to different application scenarios. Experimental evaluations show that `PIPR` outperforms various state-of-the-art systems on the binary PPI prediction problem. Moreover, it shows a promising performance on more challenging problems of interaction type prediction and binding affinity estimation, where existing approaches fall short.

- Chapter 9:

  This chapter presents a neural embedding model that captures the multi-granular associations of words and sentences based on bilingual lexicographic definitions. To enhance the learning process on limited resources, our model adopts several critical learning strategies, including multi-task learning on different bridges of languages, and joint learning of the dictionary model with a bilingual word embedding model.

Part V: Conclusion

- Chapter 10:

  The final part of the dissertation concludes the contributions of this dissertation and discusses directions for future research that can be built on top of this work.

CHAPTER 2

Background

In this chapter, we present the background on multi-relational representation learning approaches. We then describe the complex multi-relational data of different categories, each of which presents specific representation learning problems that are addressed in our work. Finally, we summarize a series of neural sequence encoding techniques, which serve as the basis of sequence data based knowledge acquisition.

2.1 Multi-relational Representation Learning

Multi-relational representation learning, a.k.a. knowledge graph embeddings, aim at distributing entities of multi-relational data in low-dimensional embedding spaces, in which the semantic similarity of entities are captured as vector distances. The relational inferences of entities are captured in forms of simple vector algebra. Models for knowledge graph embeddings can be categorized into three families, i.e. *translation-based models*, *similarity-based models*, and *neural models* (Wang et al., 2017a).

To characterize a triple (h, r, t), translation-based models follow a common assumption $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$, where $\mathbf{h}_r$ and $\mathbf{t}_r$ are either the original vectors of h and t, or the transformed vectors under a certain transformation w.r.t. relation r. The forerunner TransE (Bordes et al., 2013) sets $\mathbf{h}_r$ and $\mathbf{t}_r$ as the original $\mathbf{h}$ and $\mathbf{t}$, and achieves promising results in handling 1-to-1 relations. Later works improve TransE on multi-mapping relations by introducing relation-specific transformations on entities to obtain different $\mathbf{h}_r$ and $\mathbf{t}_r$, including projections on relation-specific hyperplanes in TransH (Wang et al., 2014b), linear transformations to heterogeneous relation spaces in TransR (Lin et al., 2015), dynamic matrices in TransD (Ji et al., 2015), and other forms (Jia et al., 2016; Nguyen et al., 2016). All these variants of TransE specialize entity embeddings for different relations, therefore improving knowledge graph completion on multi-mapping relations at the cost of increased model complexity. Meanwhile translation-based models cooperate well with other models. For example, variants of TransE are combined with word embeddings to help relation extraction from text (Weston et al., 2013; Zhong et al., 2015).

As for the similarity-based models, DistMult (Yang et al., 2015b) associates related entities using Hadamard product of embeddings, and HolE (Nickel et al., 2016) substitutes Hadamard product with circular correlation to improve the encoding of asymmetric relations, and achieves the state-of-the-art performance in knowledge graph completion. ComplEx (Trouillon et al., 2016) migrates DistMult in a complex space and offers comparable performance.

In addition to those, neural models include RESCAL (Nickel et al., 2011), SLM (Socher et al., 2013), ConvE (Dettmers et al., 2018) and R-GCN (Schlichtkrull et al., 2018). These approaches also achieve comparable performances on triple completion tasks at the cost of high model complexity.

A recent survey has summarized the score functions of these models (Wang et al., 2017a). The training process minimizes the total loss, which is defined as the sum of the scores over all triples in the graph. To prevent the training process from overfitting, negative sampling (Bordes et al., 2013) is used in training. This is realized by randomly corrupting the subject or object of a golden triple (h, r, t) to a corrupted triple (h', r, t'). Thereby the score function of a triple is described by the following hinge loss,

$$\max(f_r(\mathbf{h}, \mathbf{t}) + \gamma - f_r(\mathbf{h}', \mathbf{t}'), 0)$$

where γ is a positive margin. Usually, negative sampling follows either uniform distribution or Bernoulli distribution to corrupt either h or t, which are so called uniform and Bernoulli negative sampling respectively (Wang et al., 2014b).

It is noteworthy that the literature has paid attention only to encode simple triples within a single knowledge graph. Existing knowledge bases however, constitute far more complicated knowledge than simple relation facts, including different language specific versions, complex properties of relation facts, and knowledge of other modalities. In the next section, we describe different categories of complicated knowledge for which we seek to extend the representation learning methods towards.

2.2 Knowledge Bases and Knowledge Graphs

This section provides a introduction and categorization of the multi-relational data (or knowledge graphs).

2.2.1 Monolingual and Multilingual Knowledge

We have already come across much of monolingual knowledge in the literature. In current knowledge bases, such as Wikipedia (Wikipedia, 2016), WordNet (Bond and Foster, 2013), and ConceptNet (Speer and Havasi, 2013), vast amounts of multilingual knowledge are being created across the multiple language-specific versions of the knowledge base. Such multilingual knowledge, including inter-lingual links (ILLs), and triple-wise alignment (TWA), is very useful in aligning and synchronizing different language-specific versions of a knowledge base that evolve independently, as needed to further improve applications built on multilingual knowledge bases. However, such cross-lingual knowledge is far from complete, while extending it is challenging due to the fact that it is almost not possible for existing corpus to directly provide such knowledge of expertise. Existing approaches involve either extensive human involvement or require training comprehensive models on information that is external to knowledge graphs.

2.2.2 Ontology and Instance-level Knowledge Graphs

From a different perspective, knowledge bases can also be classified into *instance-level knowledge graphs* and *ontology-level knowledge graphs* (Ni et al., 2016). Some large knowledge bases, such as DBpedia (Lehmann et al., 2015), YAGO (Mahdisoltani et al., 2015) and ConceptNet (Speer et al., 2017), simultaneously manage both categories of knowledge graphs as two views. These two views of knowledge graphs are described as follows: (i) the **instance-level knowledge graphs** that contain **relations** between specific **entities** in triples (for example, "*Barack Obama*", "*isPoliticianOf*", "*United States*") and (ii) the **ontology-level knowledge graphs** that constitute semantic **meta-relations** of abstract **concepts** (such as "*polication*", "*is leader of*", "*city*"). In addition, these knowledge bases also provide **cross-view** links that connect ontological concepts and instances, denoting whether an instance is an instantiation from a specific concept. Figure 2.1 shows a snapshot of such a knowledge base.

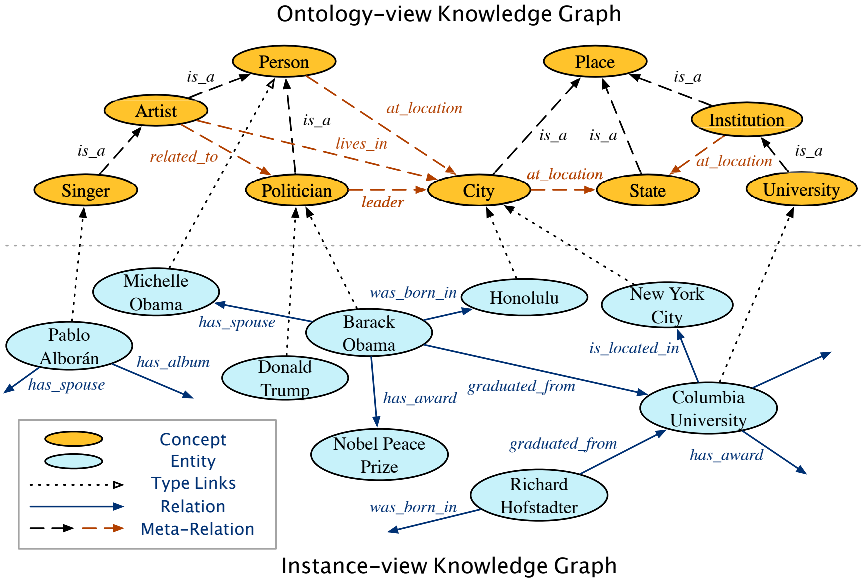


Figure 2.1: An example of two-view KB. Regular meta-relations and those conforming the hierarchical property are denoted as black and orange dashed lines respectively in the ontology view.

2.2.3 Comprehensive Properties of Relation Facts

Aside from general knowledge graphs that model relation facts as simple triples, a handful of commonsense (Mitchell et al., 2018; Speer et al., 2017) and biological ontologies (Moal and Fernández-Recio, 2012; Szklarczyk et al., 2016), feature comprehensive properties in their relation facts. This subsection describes such properties from two perspectives, i.e. *relational properties* and *uncertainty*.

Table 2.1: The number of triples of each relation type in Yago3 Ontology.

Relation	Number	Trans.	Sym.	Hier.
happenedIn	2810			
hasChild	41938			✓
hasAcademicAdvisor	4			✓
livesIn	1600			
isCitizenOf	1197			
isLocatedIn	1549685	✓		✓
wasBornIn	11672			
isMarriedTo	8593		✓	
isLeaderOf	1071			✓
isPoliticianOf	4833			
hasNeighbor	450		✓	
hasCapital	5280			
isConnectedTo	26966	✓	✓	
dealsWith	821		✓	
influences	170			
hasCurrency	4			✓
diedIn	7195			
hasGender	34811			✓
Total num/portion	1699100	92.8%	2.2%	95.8%

2.2.3.1 Relational Properties

In some knowledge graphs, especially ontology graphs, the majority of relation facts can be enforced with specific relational properties (e.g., transitivity, symmetry), or form hierarchies (e.g. taxonomy relations and spatial topological relations (Chen et al., 2016a)). For example, Freebase contains more than 20% of transitive or symmetric relations (Bollacker et al., 2008); ConceptNet (Speer and Havasi, 2013) contains 70% of transitive or symmetric relations, and at least 26% of hierarchical relations; Yago3 Ontology (Mahdisoltani et al., 2015) even contains only 17 types of relations (whose statistics we have listed in Table 2.1), while more than 92% of the relations are transitive or symmetric relations, and more than 95% of the relations are hierarchical. Moreover, we can further divide hierarchical relations into refinement and coercion relations (Camossi et al., 2006), such that the former divides each coarser concept or entity into more refined ones, and the later does the opposite.

2.2.3.2 Uncertain Knowledge Graphs

In contrast to the aforementioned deterministic knowledge graphs, uncertain knowledge graphs provide a confidence score along with every relation fact. The development of relation extraction and crowdsourcing in recent years enabled the construction of large-scale uncertain knowledge bases. ConceptNet (Speer et al., 2017) is a multilingual uncertain knowledge graph for commonsense knowledge that is collected via crowdsourcing. The confidence scores in ConceptNet mainly come from the co-occurrence frequency of the labels in crowdsourced task results. Probase (Wu et al., 2012) consists of an universal probabilistic taxonomy that is built by relation extraction. Every relation fact in Probase is associated with a joint probability $P_{isA}(x, y)$. NELL (Mitchell et al., 2018) collects relation facts from reading web pages, and learns their confidence scores from semi-supervised learning with Expectation-maximum (EM) algorithm. On the other side, in biological knowledge graphs, the confidence score often serves as a quantification of certain biochemical interactions, or express the belief of the interactions based on the experimental verification. Such cases include binding affinity estimation of proteins that are endowed to the protein-protein interactions in SKEMPI (Moal and Fernández-Recio, 2012), as well as the evidential confidence of typed protein-protein interactions in STRING (Szklarczyk et al., 2016).

2.2.4 Sequence Data As Side Information

Besides the multi-relational structures, some knowledge bases also provide side information of entities (concepts) as sequence data. Such sequence data serve as alternative views to represent entities or concepts in the embedding space, which is captured with the neural sequence models introduced in the next sections. Such data includes natural language descriptions of entities in multilingual knowledge bases (Bollacker et al., 2008; Lehmann et al., 2015), which are leveraged to support co-training of cross-lingual knowledge transfer in Section 3. Other forms include definitions of words in lexicographic knowledge bases (online dictionaries) (Meyer and Gurevych, 2012), and amino acid sequences of proteins in protein knowledge bases (Szklarczyk et al., 2016), for which we utilize directly to experiment sequence-based relational learning.

2.3 Neural Sequence Models

This subsection introduces a variety of neural sequence encoding techniques, which are the basis of our work on knowledge acquisition from sequence data.

2.3.1 The Convolution Layer with Pooling

We use $X = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_l]$ to denote an input vector sequence that corresponds to the input sequence or the outputs of a previous neural layer. A convolution layer applies a weight-sharing kernel $\mathbf{M}_c \in \mathbb{R}^{h \times k}$ to generate a k-dimension latent vector $\mathbf{h}_t^{(1)}$ from a window $\mathbf{v}_{t:t+h-1}$ of the input vector sequence X:

$$\mathbf{h}_t^{(1)} = \mathrm{Conv}(\mathbf{v}_{t:t+h-1}) = \mathbf{M}_c \mathbf{v}_{t:t+h-1} + \mathbf{b}_c$$

for which h is the kernel size, and $\mathbf{b}_c$ is a bias vector. The convolution layer applies the kernel as a sliding window to produce a sequence of latent vectors $\mathbf{H}^{(1)} = [\mathbf{h}_1^{(1)}, \mathbf{h}_2^{(1)}, ..., \mathbf{h}_{l-h+1}^{(1)}]$, where each latent vector combines the local features from each h-gram of the input sequence. The n-max-pooling mechanism is applied to every consecutive n-length subsequence (i.e., non-overlapped n-strides) of the convolution outputs, which takes the maximum value along each dimension j by $\mathbf{h}_{i,j}^{(2)} = \max(\mathbf{h}_{i:n+i-1,j}^{(1)})$. The purpose of this mechanism is to discretize the convolution results, and preserve the most significant features within each n-stride (Chen et al., 2018b; Hashemifar et al., 2018; Kim, 2014). By definition, this mechanism divides the size of processed features by n.

2.3.2 GRU and Residual GRU

The Gated Recurrent Unit model (GRU) represents an alternative to the Long-short-term Memory network (LSTM) (Cho et al., 2014), which consecutively characterizes the sequential information without using separated memory cells (Dhingra et al., 2016). Each unit consists of two types of gates to track the state of the sequence, i.e. the reset gate $\mathbf{r}_t$ and the update gate $\mathbf{z}_t$. Given the embedding $\mathbf{v}_t$ of an incoming item, GRU updates the hidden state $\mathbf{h}_t^{(3)}$ of the sequence as a linear combination of the previous state $\mathbf{h}_{t-1}^{(3)}$ and the candidate state $\tilde{\mathbf{h}}_t^{(3)}$ of a new item $\mathbf{v}_t$, which is calculated as below.

$$\begin{aligned}
&\mathbf{h}_t^{(3)} = \mathrm{GRU}(\mathbf{v}_t) = \mathbf{z}_t \odot \tilde{\mathbf{h}}_t^{(3)} + (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1}^{(3)} \\
&\mathbf{z}_t = \sigma\left(\mathbf{M}_z \mathbf{v}_t + \mathbf{N}_z \mathbf{h}_{t-1}^{(3)} + \mathbf{b}_z\right) \\
&\tilde{\mathbf{h}}_t^{(3)} = \tanh\left(\mathbf{M}_s \mathbf{v}_t + \mathbf{r}_t \odot (\mathbf{N}_s \mathbf{h}_{t-1}^{(3)}) + \mathbf{b}_s\right) \\
&\mathbf{r}_t = \sigma\left(\mathbf{M}_r \mathbf{v}_t + \mathbf{N}_r \mathbf{h}_{t-1}^{(3)} + \mathbf{b}_r\right).
\end{aligned}$$

The symbol $\odot$ denotes the element-wise multiplication. The update gate $\mathbf{z}_t$ balances the information of the previous sequence and the new item, where capitalized $\mathbf{M}_*$ and $\mathbf{N}_*$ denote different weight matrices, $\mathbf{b}_*$ denote bias vectors, and σ is the sigmoid function. The candidate state $\tilde{\mathbf{h}}_t^{(3)}$ is calculated similarly to those in a traditional recurrent unit, and the reset gate $\mathbf{r}_t$ controls how much information of the past sequence contributes to $\tilde{\mathbf{h}}_t^{(3)}$. Note that GRU generally performs comparably to LSTM in sequence encoding tasks, but is less complex and requires much fewer computational resources for training.

A *bidirectional GRU layer* characterizes the sequential information in two directions. It contains the forward encoding process $\overrightarrow{\mathrm{GRU}}$ that reads the input vector sequence $X = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_l]$ from $\mathbf{v}_1$ to $\mathbf{v}_l$, and a backward encoding process $\overleftarrow{\mathrm{GRU}}$ that reads in the opposite direction. The encoding results of both processes are concatenated for each input item $\mathbf{v}_t$, i.e. $\mathbf{h}_t^{(4)} = \mathrm{BiGRU}(\mathbf{v}_t) = [\overrightarrow{\mathrm{GRU}}(\mathbf{v}_t), \overleftarrow{\mathrm{GRU}}(\mathbf{v}_t)]$.

The *residual mechanism* passes on an identity mapping of the GRU inputs to its output side through a residual shortcut (He et al., 2016). By adding the forwarded input values to the outputs, the corresponding neural layer is only required to capture the difference between the input and output values. This mechanism aims at improving the learning process of non-linear neural layers by increasing the sensitivity of the optimization gradients (He et al., 2016; Kim et al., 2016), as well as preventing the model from the vanishing gradient problem. It has been widely deployed in deep learning architectures for various tasks of image recognition (He et al., 2016), document classification (Conneau et al., 2017b) and speech recognition (Zhang et al., 2017). In our deep RCNN in Chapter 8, the bidirectional GRU is incorporated with the residual mechanism, and will pass on the following outputs to its subsequent neural network layer:

$$\mathbf{h}_t^{(5)} = \mathrm{ResGRU}(\mathbf{v}_t) = [\overrightarrow{\mathrm{GRU}}(\mathbf{v}_t) + \mathbf{v}_t, \overleftarrow{\mathrm{GRU}}(\mathbf{v}_t) + \mathbf{v}_t]$$

In our development, we have found that the residual mechanism is able to drastically simplify the training process, and largely decreases the epochs of parameter updates for the model to converge.

2.3.3 Self-attentive Encoder

The self-attention mechanism (Conneau et al., 2017a) seeks to capture the overall meaning of the input sequence unevenly from each encoded item. One layer of self-attention is calculated as below,

$$\begin{aligned}
\mathbf{u}_t &= \tanh\left(\mathbf{M}_a \mathbf{h}_t^{(3)} + \mathbf{b}_a\right) \\
a_t &= \frac{\exp\left(\mathbf{u}_t^\top \mathbf{u}_X\right)}{\sum_{w_m \in X} \exp\left(\mathbf{u}_m^\top \mathbf{u}_X\right)} \\
\mathbf{h}_t^{(6)} &= |X| a_t \mathbf{u}_t
\end{aligned}$$

where $\mathbf{u}_t$ thereof is the intermediary latent representation of the GRU output $\mathbf{h}_t^{(3)}$, and $\mathbf{u}_X = \tanh(\mathbf{M}_a \mathbf{h}_X^{(3)} + \mathbf{b}_a)$ is the intermediary latent representation of the last GRU output $\mathbf{h}_X^{(3)}$ that can be seen as a high-level representation of the entire input sequence. By measuring the similarity of each $\mathbf{u}_t$ with $\mathbf{u}_X$, the normalized attention weight a_t for $\mathbf{h}_t^{(3)}$ is produced through a softmax function, which highlights an input that contributes more significantly to the overall meaning. Note that a scalar $|X|$ (the length of the input sequence) is multiplied along with a_t to $\mathbf{u}_t$ to obtain the weighted representation $\mathbf{h}_t^{(6)}$, so as to keep $\mathbf{h}_t^{(6)}$ from losing the original scale of $\mathbf{h}_t^{(3)}$. A latent representation of the sequence is calculated as the average of the last attention layer $E^{(2)}(X) = \frac{1}{|X|}\sum_{t=1}^{|X|} a_t \mathbf{h}_t^{(6)}$.

2.3.4 Linear Bag-of-words

The much simpler Linear Bag-of-words (BOW) encoder (Hill et al., 2016; Xie et al., 2016) is realized by the sum of projected word embeddings of the input sentence, i.e. $E^{(3)}(S) = \sum_{t=1}^{|S|} \mathbf{M}_b \mathbf{w}_t$.

CHAPTER 3

Transfer Embeddings of Multilingual Knowledge Graphs

In this chapter, we propose the first method that learns transferred embeddings across multiple knowledge graphs. This work is presented in the stage of learning to represent multilingual knowledge graphs (Chen et al., 2017b,d), although it is easily extended to other domains.

3.1 Introduction

Multilingual knowledge bases such as Wikipedia (Wikipedia, 2016), WordNet (Bond and Foster, 2013), and ConceptNet (Speer and Havasi, 2013) are becoming essential sources of knowledge for people and AI-related applications. These knowledge bases are modeled as knowledge graphs that store two aspects of knowledge: the *monolingual knowledge* that includes entities and relations recorded in the form of triples, and the *cross-lingual knowledge* that matches the monolingual knowledge among various human languages.

The coverage issue of monolingual knowledge has been widely addressed by recently proposed embedding-based techniques, which provide simple methods to encode entities in low-dimensional embedding spaces and capture relations as means of translations among entity vectors. Meanwhile, the problem of applying these techniques on cross-lingual knowledge remains largely unexplored. Such knowledge, including *inter-lingual links* (ILLs) that match the same entities, and *triple-wise alignment* (TWA) that represents the same relations, is very helpful in synchronizing different language-specific versions of a knowledge base that evolve independently, as needed to further improve applications built on knowledge bases, such as Q&A systems, semantic Web, and Web search. In spite of its importance, this cross-lingual knowledge remains largely intact. In fact, in the most successful knowledge base Wikipedia, we find that ILLs cover less than 20% entity alignment.

Leveraging knowledge graph embeddings to cross-lingual knowledge no doubt provides a generic way to help extract and apply such knowledge. However, it is a non-trivial task to find a tractable technique to capture the cross-lingual transfers. Such transfers are more difficult to capture than relational translations for several reasons: (i) a cross-lingual transfer has a far larger domain than any monolingual relational translation; (ii) it applies on both entities and relations, which have incoherent vocabularies among different languages; (iii) the known alignment for training such transfers usually accounts for a small percentage of a knowledge base. Moreover, the characterization of monolingual knowledge graph structures has to be well-preserved to ensure the correct representation of the knowledge to be aligned.

To address the above issues, we first propose a multilingual knowledge graph embedding model MTransE, that learns the multilingual knowledge graph structure using a combination of two component models, namely *knowledge model* and *alignment model*. The knowledge model encodes entities and relations in a language-specific version of knowledge graph. We explore the method that organizes each language-specific version in a separated embedding space, in which MTransE adopts TransE as the knowledge model. On top of that, the alignment model learns cross-lingual transfers for both entities and relations across different embedding spaces, where the following three representations of cross-lingual alignment are considered: distance-based axis calibration, translation vectors, and linear transformations. Thus, we obtain five variants of MTransE based on different loss functions, and identify the best variant by comparing them on cross-lingual alignment tasks using two partially aligned trilingual graphs constructed from Wikipedia triples.

While the MTransE solely relies on the structured knowledge for cross-lingual learning, it would be promising to enhance the corresponding learning process with the literal descriptions of entities that are stored in many KGs (Ji et al., 2017; Lehmann et al., 2015; Mahdisoltani et al., 2015). These descriptions comprise an alternative view of entities that potentially bridges two languages, since the descriptions of an entity in different languages often share a lot of semantic

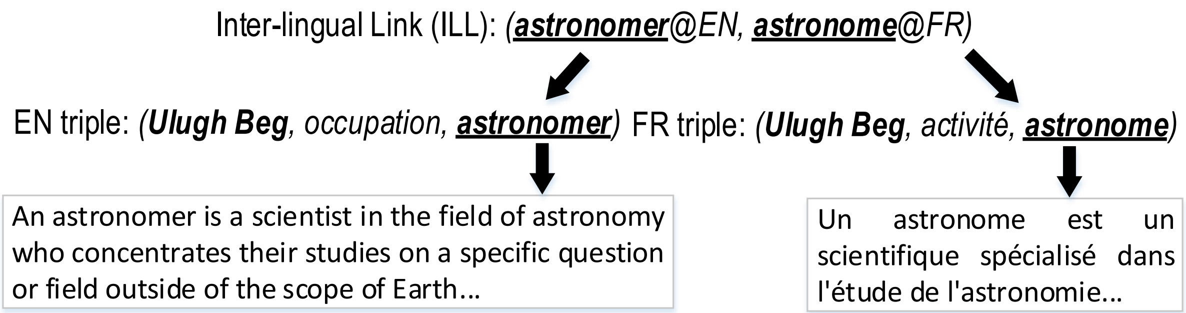

Inter-lingual Link (ILL): (astronomer@EN, astronome@FR)
EN triple: (Ulugh Beg, occupation, astronomer)
FR triple: (Ulugh Beg, activité, astronome)
An astronomer is a scientist in the field of astronomy who concentrates their studies on a specific question or field outside of the scope of Earth...
Un astronome est un scientifique spécialisé dans l'étude de l'astronomie...

Figure 3.1: A simple example which shows triples, an ILL, and entity descriptions in a multilingual KG (DBpedia). The French description for *astronome* means *an astronomer is a scientist specialized in the study of astronomy*, which contains much fewer content details than the English description for *astronomer*.

information. However, it is non-trivial to characterize and utilize such information for cross-lingual learning, as this requires the model to learn to match descriptions across different languages with inadequate labels, while conquering the inconsistency of literals in content details, grammars, and word orders (as shown in Fig. 3.1). Moreover, aggregating semantic relatedness of descriptions from words of different languages is another challenge.

To address these issues, we propose a novel co-training-based approach `KDCoE` to enhance the semi-supervised learning of multilingual KG embeddings. `KDCoE` iteratively trains two component embedding models on multilingual KG structures and entity descriptions respectively. A KG embedding model jointly trains a translational knowledge model with a linear-transformation-based alignment model to encode the KG structure. A description embedding model employs an attentive gated recurrent unit encoder (AGRU) and multilingual word embeddings to characterize multilingual entity descriptions, and is trained to collocate the embeddings of cross-lingual counterparts. The co-training is processed on a large Wikipedia-based trilingual KG, for which a very small portion of ILLs is used for training. During each iteration of co-training, both models alternately propose a set of most confident new ILLs to strengthen the supervision of cross-lingual learning, which leads to gradually improved accuracy on cross-lingual inferences. Experimental results on entity alignment confirms the effectiveness of `KDCoE` that significantly outperforms previous models, while those results on zero-shot alignment and cross-lingual KG completion also show wider usability of our approach.

3.2 Related Work

While, at the best of our knowledge, there is no previous work on learning multilingual knowledge graph embeddings, we will describe next three lines of work which are closely related to this topic.

Multilingual Word Embeddings. Several approaches learn multilingual word embeddings on parallel text corpora. Some of those can be extended to multilingual knowledge graphs, such as LM (Mikolov et al., 2013a) and CCA (Faruqui and Dyer, 2014) which induce offline transfers among pre-trained monolingual embeddings in forms of linear transformations and canonical correlation analysis respectively. These approaches do not adjust the inconsistent vector spaces via calibration or jointly training with the alignment model, thus fail to perform well on knowledge graphs as the parallelism exists only in small portions. A better approach OT (Xing et al., 2015) jointly learns regularized embeddings and orthogonal transformations, which is however found to be overcomplicated due to the inconsistency of monolingual vector spaces and the large diversity of relations among entities.

Knowledge Bases Alignment. Some projects produce cross-lingual alignment in knowledge bases at the cost of extensive human involvement and designing hand-crafted features dedicated to specific applications. Wikidata (Vrandečić, 2012) and DBpedia (Lehmann et al., 2015) rely on crowdsourcing to create ILLs and relation alignment. YAGO (Mahdisoltani et al., 2015) mines association rules on known matches, which combines many confident scores and requires extensively fine tuning. Many other works require sources that are external to the graphs, from well-established schemata or ontologies (Nguyen et al., 2011; Rinser et al., 2013; Suchanek et al., 2011) to entity descriptions (Yang et al., 2015d), which being unavailable to many knowledge bases such as YAGO, WordNet, and ConceptNet (Speer and Havasi, 2013). Such approaches also involve complicated model dependencies that are not tractable and reusable. By contrast, embedding-based methods are simple and general, require little human involvement, and generate task-independent features that can contribute to other NLP tasks.

Co-training. Co-training combines multiple models to learn on different views of the data in the training process, in which all participating models take turn in suggesting more labels on unlabeled data to enhance the supervision. This technique is widely used in semi-supervised learning tasks, such as sentiment classification on bilingual corpora with incomplete labels (Wan, 2009), collaborative filtering in recommender systems with multiple user views (Zhang et al., 2014), and semantic role labeling based on the semantic and syntactic views of documents (Thi et al., 2016). Our work conducts co-training on two views of the multilingual KG, i.e. structures and literal descriptions, which to the best of our knowledge, is the first work that incorporates co-training into embedding learning, as well as knowledge alignment tasks.

3.3 The Vanilla Multilingual Knowledge Graph Embeddings

We hereby begin our modeling with the formalization of multilingual knowledge graphs.

3.3.1 Multilingual Knowledge Graphs

In a knowledge base KB, we use $\mathcal{L}$ to denote the set of languages, and $\mathcal{L}^2$ to denote the 2-combination of $\mathcal{L}$ (i.e., the set of *unordered* language pairs). For a language $L \in \mathcal{L}$, G_L denotes the language-specific knowledge graph of L, and E_L and R_L respectively denote the corresponding vocabularies of entity expression and relation expression. $T = (h, r, t)$ denotes a triple in G_L such that $h, t \in E_L$ and $r \in R_L$. Boldfaced $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ respectively represent the embedding vectors of head h, relation r, and tail t. For a language pair $(L_1, L_2) \in \mathcal{L}^2$, $\delta(L_1, L_2)$ denotes the alignment set which contains the pairs of triples that have already been aligned between L_1 and L_2. The alignment set commonly exists in a small portion in a multilingual knowledge base (Lehmann et al., 2015; Mahdisoltani et al., 2015; Vrandečić, 2012), and is one part of knowledge we want to extend. Besides, for KBs with entity-level alignment $I(L_1, L_2)$ denotes a set of ILLs that align entities between L_1 and L_2, such that $e_1 \in E_{L_1}$ and $e_2 \in E_{L_2}$ for each ILL $(e_1, e_2) \in I(L_1, L_2)$. We assume the entity pairs have a 1-to-1 mapping and it is specified in $I(L_1, L_2)$. This assumption is congruent to the design of mainstream KGs (Lehmann et al., 2015). Besides the above structured knowledge, we use D_L to denote the literal descriptions of entities in language L. A description $d_e \in D_L$ describes an entity $e \in E_L$ with a sequence of words from the word vocabulary W_L, i.e. $d_e = \{w_1, w_2, ..., w_l\}$.

`MTransE` consists of two components that learn on the two facets of KB: the knowledge model that encodes the entities and relations from each language-specific graph structure, and the alignment model that learns the cross-lingual transfers from the existing alignment. We define a model for each language pair from $\mathcal{L}^2$ that has a non-empty alignment set. Thus, for a KB with more than two languages, a set of models composes the solution. In the following, we use a language pair $(L_i, L_j) \in \mathcal{L}^2$ as an example to describe how we define each component of a model.

3.3.2 Knowledge Model

For each language $L \in \mathcal{L}$, a dedicated k-dimensional embedding space $\mathbb{R}^k_L$ is assigned for vectors of E_L and R_L, where $\mathbb{R}$ is the field of real numbers. We adopt the basic translation-based method of TransE for each involved language, which benefits the cross-lingual tasks by representing embeddings uniformly in different contexts of relations. Therefore its loss function is given below:

$$S_K = \sum_{L \in \{L_i, L_j\}} \sum_{(h,r,t) \in G_L \wedge (\hat{h},r,\hat{t}) \notin G_L} [f_r(h,t) - f_r(\hat{h},\hat{t}) + \gamma]_+$$

for which $f_r(h,t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2$ is the dissimilarity measure of a triple (h,r,t), γ is a positive margin, $[x]_+$ denotes the positive part of x (i.e. $\max(x,0)$), and $(\hat{h},r,\hat{t})$ is a Bernoulli negative-sampled triple (Wang et al., 2014b) by substituting either h or t in (h,r,t).

3.3.3 Alignment Model

The objective of the alignment model is to construct the transfers between the vector spaces of L_i and L_j. Its loss function is given as below:

$$S_A = \sum_{(T,T') \in \delta(L_i,L_j)} S_a(T,T')$$

for which the alignment score $S_a(T,T')$ iterates through all pairs of aligned triples. Three different techniques to score the alignment are considered: distance-based axis calibration, translation vectors, and linear transformations. Each of them is based on a different assumption, and constitutes different forms of S_a alongside.

Distance-based Axis Calibration. This type of alignment models penalize the alignment based on the distances of cross-lingual counterparts. Either of the following two scorings can be adopted to the model.

$$S_{a_1} = \|\mathbf{h} - \mathbf{h}'\| + \|\mathbf{t} - \mathbf{t}'\|$$

S_{a_1} regulates that correctly aligned multilingual expressions of the same entity tend to have close embedding vectors. Thus by minimizing the loss function that involves S_{a_1} on known pairs of aligned triples, the alignment model adjusts axes of embedding spaces towards the goal of coinciding the vectors of the same entity in different languages.

$$S_{a_2} = \|\mathbf{h} - \mathbf{h}'\| + \|\mathbf{r} - \mathbf{r}'\| + \|\mathbf{t} - \mathbf{t}'\|$$

S_{a_2} overlays the penalty of relation alignment to S_{a_1} to explicitly converge coordinates of the same relation.

The alignment models based on axis calibration assume analogous spatial emergence of items in each language. Therefore, it realizes the cross-lingual transfer by carrying forward the vector of a given entity or relation from the space of the original language to that of the other language.

Translation Vectors. This model encodes cross-lingual transfers into vectors. It consolidates alignment into graph structures and characterizes cross-lingual transfers as regular relational translations. Hence S_{a_3} as below is derived.

$$S_{a_3} = \left\|\mathbf{h} + \mathbf{v}^e_{ij} - \mathbf{h}'\right\| + \left\|\mathbf{r} + \mathbf{v}^r_{ij} - \mathbf{r}'\right\| + \left\|\mathbf{t} + \mathbf{v}^e_{ij} - \mathbf{t}'\right\|$$

$\mathbf{v}^e_{ij}$ and $\mathbf{v}^r_{ij}$ thereof are respectively deployed as the entity-dedicated and relation-dedicated translation vectors between L_i and L_j, such that we have $\mathbf{e} + \mathbf{v}^e_{ij} \approx \mathbf{e}'$ for embedding vectors $\mathbf{e}$, $\mathbf{e}'$ of the same entity e expressed in both languages, and $\mathbf{r} + \mathbf{v}^r_{ij} \approx \mathbf{r}'$ for those of the same relation. We deploy two translation vectors instead of one, because there are far more distinct entities than relations, and using one vector easily leads to imbalanced signals from relations.

Such a model obtains a cross-lingual transfer of an embedding vector by adding the corresponding translation vector. Moreover, it is easy to see that $\mathbf{v}^e_{ij} = -\mathbf{v}^e_{ji}$ and $\mathbf{v}^r_{ij} = -\mathbf{v}^r_{ji}$ hold. Therefore, as we obtain the translation vectors from L_i to L_j, we can always use the same vectors to translate in the opposite direction.

Linear Transformations. The last category of alignment models deduce linear transformations between embedding spaces. S_{a_4} as below learns a $k \times k$ square matrix $\mathbf{M}^e_{ij}$ as a linear transformation on entity vectors from L_i to L_j, given k as the dimensionality of the embedding spaces.

$$S_{a_4} = \left\|\mathbf{M}^e_{ij}\mathbf{h} - \mathbf{h}'\right\| + \left\|\mathbf{M}^e_{ij}\mathbf{t} - \mathbf{t}'\right\|$$

S_{a_5} additionally brings in a second linear transformation $\mathbf{M}^r_{ij}$ for relation vectors, which is of the same shape as $\mathbf{M}^e_{ij}$. The use of a different matrix is again due to different redundancy of entities and relations.

$$S_{a_5} = \left\|\mathbf{M}^e_{ij}\mathbf{h} - \mathbf{h}'\right\| + \left\|\mathbf{M}^r_{ij}\mathbf{r} - \mathbf{r}'\right\| + \left\|\mathbf{M}^e_{ij}\mathbf{t} - \mathbf{t}'\right\|$$

Unlike axis calibration, linear-transformation-based alignment model treats cross-lingual transfers as the topological transformation of embedding spaces without assuming the similarity of spatial emergence.

The cross-lingual transfer of a vector is obtained by applying the corresponding linear transformation. It is noteworthy that, regularization of embedding vectors in the training process (which will be introduced soon after) ensures the invertibility of the linear transformations such that ${\mathbf{M}^e_{ij}}^{-1} = \mathbf{M}^e_{ji}$ and ${\mathbf{M}^r_{ij}}^{-1} = \mathbf{M}^r_{ji}$. Thus the transfer in the revert direction is always enabled even though the model only learns the transformations of one direction.

3.3.4 Variants of `MTransE`

Combining the above two component models, `MTransE` minimizes the following loss function $J = S_K + \alpha S_A$, where α is a hyperparameter that weights S_K and S_A.

As we have given out five variants of the alignment model, each of which correspondingly defines its specific way of computing cross-lingual transfers of embedding vectors. We denote Var_k as the variant of `MTransE` that adopts the k-th alignment model which employs S_{a_k}. In practice,

Table 3.1: Summary of model variants.

Var	Model Complexity	Cross-lingual transfer	Search Complexity
Var$_1$	$\mathrm{O}(n_e kl + n_r kl)$	$\tau_{ij}(\mathbf{e}) = \mathbf{e}$ $\tau_{ij}(\mathbf{r}) = \mathbf{r}$	$\mathrm{O}(n_e k)$ $\mathrm{O}(n_r k)$
Var$_2$	$\mathrm{O}(n_e kl + n_r kl)$	$\tau_{ij}(\mathbf{e}) = \mathbf{e}$ $\tau_{ij}(\mathbf{r}) = \mathbf{r}$	$\mathrm{O}(n_e k)$ $\mathrm{O}(n_r k)$
Var$_3$	$\mathrm{O}(n_e kl + n_r kl$ $+ kl^2)$	$\tau_{ij}(\mathbf{e}) = \mathbf{e} + \mathbf{v}_{ij}^e$ $\tau_{ij}(\mathbf{r}) = \mathbf{r} + \mathbf{v}_{ij}^r$	$\mathrm{O}(n_e k)$ $\mathrm{O}(n_r k)$
Var$_4$	$\mathrm{O}(n_e kl + n_r kl$ $+ 0.5k^2 l^2)$	$\tau_{ij}(\mathbf{e}) = \mathbf{M}_{ij}^e \mathbf{e}$ $\tau_{ij}(\mathbf{r}) = \mathbf{M}_{ij}^e \mathbf{r}$	$\mathrm{O}(n_e k^2 + n_e k)$ $\mathrm{O}(n_r k^2 + n_r k)$
Var$_5$	$\mathrm{O}(n_e kl + n_r kl$ $+ k^2 l^2)$	$\tau_{ij}(\mathbf{e}) = \mathbf{M}_{ij}^e \mathbf{e}$ $\tau_{ij}(\mathbf{r}) = \mathbf{M}_{ij}^r \mathbf{r}$	$\mathrm{O}(n_e k^2 + n_e k)$ $\mathrm{O}(n_r k^2 + n_r k)$

Notation: $\mathbf{e}$ and $\mathbf{r}$ are respectively the vectors of an entity e and a relation r, k is the dimension of the embedding spaces, l is the cardinality of $\mathcal{L}$, n_e and n_r are respectively the number of entities and the number of relations, where $n_e \gg n_r$.

the searching of a cross-lingual counterpart for a source is always done by querying the nearest neighbor from the result point of the cross-lingual transfer. We denote function τ_{ij} that maps a cross-lingual transfer of a vector from L_i to L_j, or simply τ in a bilingual context. As stated, the solution in a multi-lingual scenario consists of a set of models of the same variant defined on every language pair in $\mathcal{L}^2$. Table 3.1 summarizes the model complexity, the definition of cross-lingual transfers, and the complexity of searching a cross-lingual counterpart for each variant.

We optimize the loss function using on-line stochastic gradient descent (Wilson and Martinez, 2003). At each step, we update the parameter θ by setting $\theta \leftarrow \theta - \lambda \nabla_\theta J$, where λ is the learning rate. Instead of directly updating J, our implementation optimizes S_K and αS_A alternately. In detail, at each epoch we optimize $\theta \leftarrow \theta - \lambda \nabla_\theta S_K$ and $\theta \leftarrow \theta - \lambda \nabla_\theta \alpha S_A$ in separated groups of steps.

We enforce the constraint that the l_2 norm of any entity embedding vector is 1, thus regularize embedding vectors to a unit spherical surface. This constraint is employed in the literature (Bordes et al., 2013, 2014b; Jenatton et al., 2012) and has two important effects: (i) it helps avoid the case where the training process trivially minimizes the loss function by shrinking the norm of embedding vectors, and (ii) it implies the invertibility of the linear transformations (Xing et al., 2015) for Var$_4$ and Var$_5$.

Table 3.2: Statistics of the WK3l datasets.

dataset	#En triples	#Fr triples	#De triples	#Aligned triples
WK3l-15k	203,502	170,605	145,616	En-Fr:16,470 En-De:37,170
WK3l-120k	1,376,011	767,750	391,108	En-Fr:124,433 En-De:69,413

Table 3.3: Number of entity inter-lingual links (ILLs).

Dataset	En-Fr	Fr-En	En-De	De-En
WK3l-15k	3,733	3,815	1,840	1,610
WK3l-120k	42,413	41,513	7,567	5,921

We initialize vectors by drawing from a uniform distribution on the unit spherical surface, and initialize matrices using random orthogonal initialization (Saxe et al., 2014).

3.4 Cross-lingual Entity Matching

In this section, we evaluate the proposed methods on cross-lingual entity matching.

Datasets. Experimental results on the trilingual datasets WK3l are reported in this section. WK3l contains English (En), French (Fr), and German (De) knowledge graphs under DBpedia's `dbo:Person` domain, where a part of triples are aligned by verifying the ILLs on entities, and multilingual labels of the DBpedia ontology on some relations. The number of entities in each language is adjusted to obtain two datasets. For each of the three languages thereof, WK3l-15k matches the number of nodes (about 15,000) with FB15k—the largest monolingual graph used by many recent works (Ji et al., 2015; Jia et al., 2016; Lin et al., 2015; Zhong et al., 2015), and the number of nodes in WK3l-120k is several times larger. For both datasets, German graphs are sparser than English and French graphs. We also collect extra entity ILLs for the evaluation of cross-lingual entity matching, whose quantity is shown in Table 3.3.

The objective of this task is to match the same entities from different languages in KB. Due to the large candidate space, this task emphasizes more on ranking a set of candidates rather than acquiring the best answer. We perform this task on both datasets to compare five variants of `MTransE`.

Table 3.4: Cross-lingual entity matching results.

Dataset	WK3l-15k								WK3l-120k			
Aligned Languages	En-Fr		Fr-En		En-De		De-En		En-Fr	Fr-En	En-De	De-En
Metric	Hits@10	Mean	Hits@10	Mean	Hits@10	Mean	Hits@10	Mean	Hits@10	Hits@10	Hits@10	Hits@10
LM	12.31	3621.17	10.42	3660.98	22.17	5891.13	15.21	6114.08	11.74	14.26	24.52	13.58
CCA	20.78	3094.25	19.44	3017.90	26.46	5550.89	22.30	5855.61	19.47	12.85	25.54	20.39
OT	44.97	508.39	40.92	461.18	44.47	155.47	49.24	145.47	38.91	37.19	38.85	34.21
Var_1	51.05	470.29	46.64	436.47	48.67	146.13	50.60	167.02	38.58	36.52	42.06	47.79
Var_2	45.25	570.72	41.74	565.38	46.27	168.33	49.00	211.94	31.88	30.84	41.22	40.39
Var_3	38.64	587.46	36.44	464.64	50.82	125.15	52.16	151.84	38.26	36.45	50.48	52.24
Var_4	59.24	**190.26**	**57.48**	**199.64**	**66.25**	**74.62**	**68.53**	**42.31**	**48.66**	47.43	57.56	63.49
Var_5	**59.52**	191.36	57.07	204.45	60.25	99.48	66.03	54.69	45.65	**47.48**	**64.22**	**67.85**

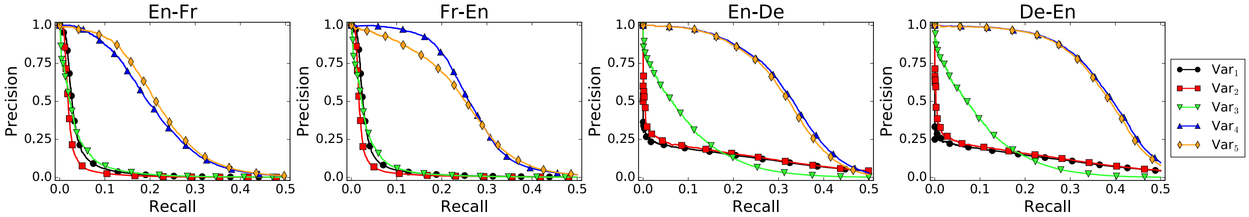


Figure 3.2: Precision-recall curves for cross-lingual entity matching on WK3l-15k.

To show the superiority of `MTransE`, we adapt LM, CCA, and OT (which are introduced in Section 3.2) to their knowledge graph equivalences.

Evaluation Protocol. Each `MTransE` variant is trained on a complete dataset. LM and CCA are implemented by inducing the corresponding transformations across separately trained knowledge models on monolingual graphs, while using the alignment sets as anchors. Training OT is quite similar to `MTransE`, we add the process of orthogonalization to the training of the alignment model, since the regularization of vectors has already been enforced. The entity ILLs are used as ground truth for test. We take these unidirectional links between English-French and English-German, i.e., four directions in total. For each ILL (e, e'), we perform a kNN search from the cross-lingual transfer point of e (i.e., $\tau(\mathbf{e})$) and record the rank of $\mathbf{e}'$. Following the convention (Jia et al., 2016; Xing et al., 2015), we aggregate two metrics over all test cases, i.e., the proportion of ranks no larger than 10 $Hits@10$ (in percentage), and the mean rank $Mean$. We prefer higher $Hits@10$ and lower $Mean$ that indicate a better outcome.

For training, we select the learning rate λ among {0.001, 0.01, 0.1}, α among {1, 2.5, 5, 7.5}, l_1 or l_2 norm in loss functions, and dimensionality k among {50, 75, 100, 125}. The best configuration on WK3l-15k is $\lambda = 0.01$, $\alpha = 5$, $k = 75$, l_1 norm for Var_1, Var_2, LM, and CCA, l_2 norm for other variants and OT. While the best configuration on WK3l-120k is $\lambda = 0.01$, $\alpha = 5$,

$k = 100$, and l_2 norm for all models. The training on both datasets takes 400 epochs.

Results. We report $Hits@10$ and $Mean$ for WK3l-15k, and $Hits@10$ for WK3l-120k, on the four involved directions of cross-lingual matching in Table 3.4. As expected, without jointly adapting the monolingual vector spaces with the knowledge alignment, LM and CCA are largely outperformed by the rest. While the orthogonality constraint being too strong to be enforced in these cases, OT performs at most closely to the simplest cases of `MTransE`. For `MTransE`, Var$_4$ and Var$_5$ outperform the other three variants under all settings. The fairly close results obtained by these two variants indicate that the interference caused by learning an additional relation-dedicated transformation in Var$_5$ is negligible to the entity-dedicated transformation. Correspondingly, we believe that the reason for Var$_3$ to be outperformed by Var$_4$ and Var$_5$ is that it fails to differentiate well the over-frequent cross-lingual alignment from regular relations. Therefore, the characterization for cross-lingual alignment is negatively affected by the learning process for monolingual relations in a visible degree. Axis calibration appears to be unstable on this task. We hypothesize that this simple technique is affected by two factors: coherence between language-specific versions, and density of the graphs. Var$_2$ is always outperformed by Var$_1$ due to the negative effect of the calibration based on relations. We believe this is because multi-mapping relations are not so well-captured by TransE as explained in (Wang et al., 2014b), therefore disturb the calibration of the entire embedding spaces. Although Var$_1$ still outperforms Var$_3$ on entity matching between English and French graphs in WK3l-15k, coherence somewhat drops alongside when scaling up to the larger dataset so as to hinder the calibration. The German graphs are sparse, thus should have set a barrier for precisely constructing embedding vectors and hindered calibration on the other side. Therefore Var$_1$ still performs closely to Var$_3$ in the English-German task on WK3l-15k and English-French task on WK3l-120k, but is outperformed by Var$_3$ in the last setting. In general, the variants that use linear transformations are the most desired. This conclusion is supported by their promising outcome on this task, and it is also reflected in the precision-recall curves shown in Figure 3.2.

3.4.1 Examples of Knowledge Alignment

We have already shown the effectiveness of MTransE in aligning cross-lingual knowledge, especially the linear-transformation-based variants Var_4 and Var_5. In this part we illustrate our methods with some examples in order to reveal insights on how our methods may be used in cross-lingual knowledge augmentation.

Table 3.5: Examples of cross-lingual entity matching.

Entity	Target	Candidates (in ascending order of rank by Euclidean distance)
Barack Obama	French	**Barack Obama**, *George Bush*, *Jimmy Carter*, George Kalkoa
	German	**Barack Obama**, *Bill Clinton*, *George h. w. Bush*, Hamid Karzai
Paris	French	**Paris**, *Amsterdam*, **à Paris**, *Manchester*, De Smet
	German	**Paris**, *Languedoc*, *Constantine*, *Saint-maurice*, *Nancy*
California	French	*San Francisco*, *Los Angeles*, *Santa Monica*, **Californie**
	German	**Kalifornien**, *Los Angeles*, *Palm Springs*, *Santa Monica*
rock music	French	*post-punk*, **rock alternatif**, *smooth jazz*, *soul jazz*
	German	**rockmusik**, *soul*, *death metal*, *dance-pop*

Table 3.6: Examples of cross-lingual relation matching.

Relation	Target	Candidates (in ascending order of rank by Euclidean distance)
capital	French	**capitale**, *territoire*, pays accrèditant, lieu de veneration
	German	**hauptstadt**, *hauptort*, *gründungsort*, *city*
nationality	French	**nationalié**, **pays de naissance**, *domicile*, *résidence*
	German	**nationalität**, **nation**, letzter start, *sterbeort*
language	French	**langue**, réalisations, lieu deces, *nationalitè*
	German	**sprache**, **originalsprache**, **lang**, *land*
nickname	French	**surnom**, descendant, texte, *nom de ring*
	German	**spitzname**, originaltitel, *names*, **alternativnamen**

We start with the search of cross-lingual counterparts of entities and relations. We choose an entity (or relation) in English and then show the nearest candidates in French and German, respectively. These candidates are listed by decreasing values of the Euclidean distance between their vectors in the target language space and the result point of cross-lingual transfer. Several samples are shown in Table 3.5. We also show some examples of cross-lingual relation matching in Table 3.6. , which is identical to entity matching, except that we are dealing with relations. Several examples are shown in Table 3.5 and Table 3.6. In all tables of this subsection, we mark the exact answers as **boldfaced**, and the conceptually close ones as *italic*. For example, in Table 3.5, besides

Table 3.7: Examples of cross-lingual triple completion.

Query	Target	Candidates (in ascending order of rank)
(Adam Lambert, genre, ?t)	French	*musique indèpendante*, **musique alternative**, ode, **glam rock**
	German	**popmusik**, **dance-pop**, no wave, *soul*
(Ronaldinho, position, ?t)	French	**milieu offensif**, **attaquant**, *quarterback*, *latèral gauche*
	German	**stürmer**, *linker flügel*, **angriffsspieler**, *rechter flÂĺÂźgel*
(Italy, ?r, Rome)	French	**capitale**, **plus grande ville**, **chef-lieu**, garnison
	German	**hauptstadt**, **hauptort**, verwaltungssitz, stadion
(Barack Obama, ?r, George Bush)	French	*ministre-prèsident*, **prèdècesseur**, *premier ministre*, *prèsident du conseil*
	German	**vorgänger**, **vorgängerin**, besetzung, lied
(?h, instrument, guitar)	French	**Brant Bjork**, **Chris Garneau**, *David Draiman*, **Ian Mackaye**
	German	**Phil Manzanera**, *Styles P.*, *Tina Charles*, **Luke Bryan**

boldfacing the exactly correct answers for Barack Obama and Paris, we consider those who have also been U.S. presidents as conceptually close to Barack Obama, and European cities other than Paris as conceptually close to Paris. Also, in Table 3.6, those French and German relations that have the meaning of settlements of significance are considered as conceptually close to *capital*.

We then move on to the more complicated cross-lingual triple completion task. We construct queries by replacing one element in an English triple with a question mark, for which we seek for answers in another language. Our methods need to transfer the remaining elements to the space of the target language and pick the best answer for the missing element. Table 3.7 shows some query answers. It is noteworthy that the basic queries are already useful for aided cross-lingual augmentation of knowledge. However, developing a joint model to support complex queries on multilingual knowledge graphs based on `MTransE` generated features appears to be a promising future work to support Q&A on multilingual knowledge bases.

Figure 3.3 shows the PCA projection of the *same* six English entities in their original English space and in French space after transformation. We can observe that the vectors of English entities show certain structures, where the U.S. cities are grouped together and other countries' cities are well separated. After transformation into French space, these English entities not only keep their original spatial emergence, but also are close to their corresponding entities in French. This illustrates the transformation preserves mono-lingual structure and also it is able to capture cross-

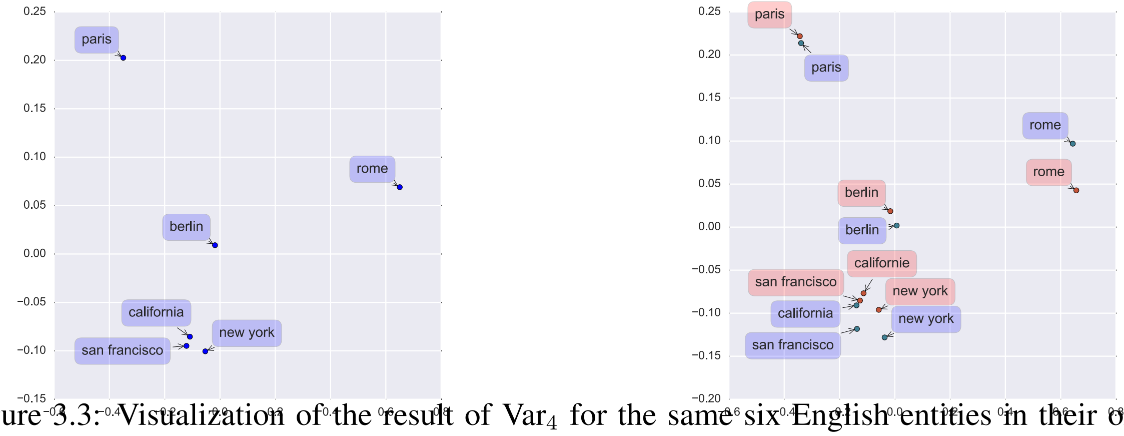


Figure 3.3: Visualization of the result of Var$_4$ for the same six English entities in their original space (left) and in French space after being transformed (right). English entities are rendered in blue, and the corresponding French entities are in light ruby.

lingual information. We believe this example illustrates the good performance we have demonstrated in cross-lingual tasks including cross-lingual entity matching and triple-wise alignment verification.

3.5 Semi-supervised Co-training

The `KDCoE` model conducts iterative co-training of two components, i.e. the multilingual KG embedding model (KGEM) and the multilingual description embedding model (DEM), which capture embeddings with cross-lingual inferences for structured knowledge and entity descriptions respectively. During co-training, both components are trained in turns to propose new ILLs with high confidence, which populate the training set and become visible to future turns of training. For KGEM, `KDCoE` adopts the `MTransE`-Var4 model based on entity-level alignment.

The DEM learns in two stages. An attentive gated recurrent unit encoder (AGRU) is used to encode the multilingual entity descriptions. On top of that, DEM is trained to collocate the description embeddings of cross-lingual counterparts. To better reflect the semantic information of multilingual entity descriptions from the word level, we use multilingual word embeddings that are capable of collocating similar words in different languages. In detail, we pre-train the cross-lingual Bilbowa (Gouws et al., 2015) word embeddings on the cross-lingual parallel corpora Europarl v7 (Koehn, 2005) and monolingual corpora of Wikipedia dump. After the pre-training, we fix the word embeddings to convert each entity description d_e to a sequence of vectors to be fed into the description encoder.

Learning Objective. We utilize an encoder of two stacked attentive GRU layers to model the descriptions of both languages, which takes the description sequence d_e and produces the embedding from the second-layer outputs. In detail, we apply an affine layer to map the averaged second-layer outputs to a common embedding space for descriptions: $\mathbf{d}_e = \tanh\left(\mathbf{M}_d\left(\frac{1}{|d_e|}\sum_{i=1}^{|d_e|}\mathbf{v}_i^{(DEM)}\right) + \mathbf{b}_d\right)$. We use the same dimensionality (denoted as k_2) for the output vectors of the second GRU layer $\mathbf{v}_i^{(2)}$ and the description embeddings $\mathbf{d}_e$. Like KG embeddings, we regularize each $\mathbf{d}_e$ as $\|\mathbf{d}_e\|_2 = 1$.

The learning objective of DEM is to maximize the log likelihood of each entity given its cross-lingual counterpart in terms of their description embeddings, which is realized by minimizing the following objective function,

$$\begin{aligned} S_D &= \sum_{(e,e')\in I(L_i,L_j)} -LL_1 - LL_2 \\ &= \sum_{(e,e')\in I(L_i,L_j)} -\log\left(P(e|e')\right) - \log\left(P(e'|e)\right) \end{aligned}$$

Similar to (Mikolov et al., 2013c), we adopt negative sampling to obtain the following computationally efficient terms of approximation for LL_1 and LL_2, where $|B_d|$ is the batched sampling size, and U is the distribution of entities.

$$\begin{aligned} LL_1 &= \log\sigma\left(\mathbf{d}_e^\top \mathbf{d}_{e'}\right) + \sum_{k=1}^{|B_d|} \mathbb{E}_{e_k\sim \mathrm{U}\left(e_k\in E_{L_i}\right)}\left[\log\sigma\left(-\mathbf{d}_{e_k}^\top \mathbf{d}_{e'}\right)\right] \\ LL_2 &= \log\sigma\left(\mathbf{d}_e^\top \mathbf{d}_{e'}\right) + \sum_{k=1}^{|B_d|} \mathbb{E}_{e_k\sim \mathrm{U}\left(e_k\in E_{L_j}\right)}\left[\log\sigma\left(-\mathbf{d}_e^\top \mathbf{d}_{e_k}\right)\right] \end{aligned}$$

Through optimization of S_D, the encoder is trained towards the goal of maximizing the dot product of each description embedding $\mathbf{d}_e$ and that of its cross-lingual counterpart $\mathbf{d}_{e'}$, and decreasing the dot product of unrelated description embeddings. Since description embeddings are regularized to unit vectors, this process is equivalent to minimizing the l_2-distance between each pair of cross-lingual counterparts (i.e. collocating). To facilitate the sampling-based approximation, we use the

Algorithm 1: Iterative co-training of KDCoE.

```
Input: Graphs $G_{L_i}, G_{L_j}$, descriptions $D_{L_i}, D_{L_j}$, ILL training set $I_{tr}$, ILL validation set $I_{val}$, candidate entities without ILLs
       $\tilde{E_{L_i}} \in E_{L_i}, \tilde{E_{L_j}} \in E_{L_j}$, precision threshold $\tau$ on $I_{val}$ for selecting proposed ILLs.
Output: parameters $\theta$ for KGEM and DEM
while Either KGEM or DEM does not propose more ILLs do
    Reinitialize KGEM and DEM;
    Train KGEM on $I_{tr}, G_{L_i}, G_{L_j}$ until $S_{KG}$ no longer improves on graphs and $I_{val}$;
    Select max $l_2$ threshold $\delta_1$, for which the precision of the predictions $(e, \hat{e'})$ by KGEM on $I_{val}$ s.t. $\left\|\mathbf{M}_{ij}\mathbf{e} - \hat{\mathbf{e}'}\right\|_2 < \delta_1$ is higher
      than $\tau$;
    for $e \in \tilde{E_{L_i}}$ do
        $\hat{\mathbf{e}'} \leftarrow \mathrm{NearestNeighbor}(\mathbf{M}_{ij}\mathbf{e}, L_j)$;                /* NN in $L_j$.  */
        if $\left\|\mathbf{M}_{ij}\mathbf{e} - \hat{\mathbf{e}'}\right\|_2 < \delta_1$ then
            $I_{tr} \leftarrow I_{tr} \cup \{(e, \hat{e'})\}$;                               /* Propose an ILL. */
            $\tilde{E_{L_i}} \leftarrow \tilde{E_{L_i}} - \{e\}; \tilde{E_{L_j}} \leftarrow \tilde{E_{L_j}} - \{\hat{e'}\}$;
    Train DEM on $I_{tr}, D_{L_i}, D_{L_j}$ until $S_D$ no longer improves on $I_{val}$;
    Select max $l_2$ threshold $\delta_2$, for which the precision of the predictions $(e, \hat{e'})$ by DEM on $I_{val}$ s.t. $\left\|\mathbf{d}_e - \mathbf{d}_{\hat{e'}}\right\|_2 < \delta_2$ is higher than
      $\tau$;
    for $e \in \tilde{E_{L_i}}$ do
        $\mathbf{d}_{\hat{e'}} \leftarrow \mathrm{NearestNeighbor}(\mathbf{d}_e, L_j)$;                  /* NN in $L_j$.  */
        if $\left\|\mathbf{d}_e - \mathbf{d}_{\hat{e'}}\right\|_2 < \delta_2$ then
            $I_{tr} \leftarrow I_{tr} \cup \{(e, \hat{e'})\}$;                               /* Propose an ILL. */
            $\tilde{E_{L_i}} \leftarrow \tilde{E_{L_i}} - \{e\}; \tilde{E_{L_j}} \leftarrow \tilde{E_{L_j}} - \{\hat{e'}\}$;
```

stratified negative sharing technique (Chen et al., 2017e). That is to say, we sample batches of ILLs into B_d. Then based on the 1-to-1 mapping of ILLs, we select negative samples for each e as all entities e_k in the other language from B_d, except for the one that forms the ILL with e.

Note that we have also explored with other forms of description encoders. Linear BOW and CNN used in (Ji et al., 2017) to represent monolingual entity descriptions fail to accurately match cross-lingual counterparts by losing the sequential and attentive information. Attentive LSTM encoders perform comparably to AGRU, but are more complex and require more computational resources for training. Adopting bidirectional encoders hinders the performance of our tasks.

3.5.1 Iterative Co-training

The co-training of the two model components is conducted iteratively on the KG, where a small amount of ILLs is provided for training. At each iteration, the component models alternately take turns of the train-and-propose process. In each turn, the model is first initialized using orthogonal initialization, and optimized using SGD with early-stopping based on a small validation set of ILLs. After training, that model predicts new ILLs for candidate entities that are not involved in any previous ILL. Such a prediction is based on a distance-based strategy, where a new ILL sourced from L_i is suggested by searching the nearest neighbor (NN) within the candidate space of L_j from the transformed entity vector, or from the original description vector. As lower l_2-distances imply more precise inferences of embeddings (Chen et al., 2017c; Mikolov et al., 2013c; Zhu et al., 2017), only the most confident predictions, for which the l_2-distance between the source and the NN falls within a certain threshold, are populated into the training set. The l_2-distance threshold is selected to ensure the prediction precision on the validation set to be above τ, so as to ensure a high estimated precision of proposed new ILLs. Both components repeatedly conduct the above train-and-propose processes, therefore gradually enhance the supervision of cross-lingual learning for each other, until either of the two model components no longer proposes new ILLs. The detailed co-training procedure of KDCoE is given in Algorithm 1.

3.6 Semi-supervised and Zero-shot Entity Alignment

We evaluate KDCoE on two knowledge alignment tasks: cross-lingual entity alignment and zero-shot alignment.

Dataset. Experiments are conducted on the trilingual dataset WK3l60k, which is extracted from the subset of DBpedia that is highly covered by ILLs in the purpose of providing enough ground truth to evaluate the semi-supervised cross-lingual learning. Statistics of the dataset is given in Table 3.8. Each language-specific version of the KG consists of 54k to 65k entities, and varies in density, which indicates the dataset to be challenging in terms of cross-lingual inconsistency and providing much larger candidate spaces than other datasets for KG embeddings that typically searches around 15k-40k entities (Sun et al., 2017b; Yang et al., 2015a). Literal descriptions covers 82%-96% of entities in each language. We extract ILLs between English-French and English-German to train and evaluate cross-lingual entity alignment, for which we use about 20% for training, 70% for testing, and the rest for validation. The proportion used for training is in accord with the estimated global completeness of ILLs in the KB (Lehmann et al., 2015). Meanwhile, another small set of entities with ILLs and descriptions are extracted, but are excluded from the KG structure for evaluating zero-shot alignment.

3.6.1 Cross-lingual Entity Alignment

The objective of this task is to match the same entities from different languages in KB. The baselines we compare against include three `MTransE` variants that adopt different alignment techniques to model ILLs, and ITransE which employs parameter sharing for self-training. We also adapt LM, CCA, and OT (as introduced in Section 2) to their KG equivalences.

Evaluation Protocol. The `MTransE` variants, ITransE, and KGEM of `KDCoE` are trained on the complete KG structures of two languages and the small training set of ILLs. LM and CCA are implemented by inducing the corresponding transformations across separately trained knowledge models. OT is implemented by enforcing `MTransE`-LT with an orthogonality constraint. DEM of `KDCoE` is trained on the entity descriptions that are covered by the current I_{tr} during each iteration of co-training. For each ILL (e, e'), the prediction is performed by a kNN search from the cross-lingual conversion point of $\mathbf{e}$, and record the rank of $\mathbf{e}'$ within related entities in the target language. Following the convention (Nickel et al., 2016), we aggregate three metrics on test cases: the accuracy $Hit@1$ (%), the proportion of ranks no larger than 10 $Hit@10$ (%), and mean reciprocal rank MRR. All three metrics are preferred to be higher to indicate better performance.

Model configuration is based on the validation set. We search the learning rate λ_1 for KGEM and other baselines among $\{0.001, 0.005, 0.01\}$, dimensionality k_1 in $\{50, 75, 100\}$, margin γ in $\{0.5, 1, 2\}$, and α in $\{1, 2.5, 5, 7.5\}$. For ITransE, we select the distance threshold δ_i for self-training among $\{0.5, 0.75, 1\}$. For DEM of `KDCoE` we select the learning rate λ_2 among $\{0.001, 0.005, 0.01\}$, dimensionality k_2 in $\{50, 75, 100\}$. We fix the batch sizes $|B_t|$ for KGEM and other models, and $|B_d|$ for DEM as 1024. The best configuration is $\lambda_1 = 0.005$, $k_1 = 50$, $\gamma = 1$, $\alpha = 2.5$, $\delta_i = 0.75$ for all KG embedding models, and $\lambda_2 = 0.001$, $k_2 = 75$ for DEM. For ILL proposing, we set the precision threshold τ to 0.9. We pre-train Bilbowa based on the setting in (Gouws et al., 2015) to obtain 200-dimensional word embeddings. The multilingual entity descriptions are delimited to the first two sentences, so as to reduce some inconsistent content details. We also remove the stop words in these descriptions, zero-pad short ones and truncate long ones to the average sequence length of 36. Training of models is always terminated via early-stopping, and the co-training pro-

Data	#En	#Fr	#De	ILL Lang	#Train	#Valid	#Test	#Zero-shot
Triples	569,393	258,337	224,647	En-Fr	13,050	2,000	39,155	5,000
Desc.	67,314	45,842	43,559	En-De	12,505	2,000	41,018	5,632

Table 3.8: Statistics of the Wk3l60k dataset.

Language	En-Fr			En-De		
Metric	$Hit@1$	$Hit@10$	MRR	$Hit@1$	$Hit@10$	MRR
LM	1.02	2.21	0.014	1.37	2.14	0.015
CCA	1.80	3.54	0.021	2.19	3.42	0.025
OT	20.15	25.37	0.212	11.04	19.74	0.122
ITransE	10.14	11.59	0.106	6.55	11.44	0.076
MTransE-AC	4.49	8.67	0.051	5.56	8.50	0.060
MTransE-TV	5.12	7.55	0.055	3.62	8.12	0.053
MTransE-LT	27.40	33.98	0.309	17.90	31.59	0.225
KDCoE ($i2$)	37.70	45.01	0.405	29.80	41.66	0.322
KDCoE ($i3$)	43.77	53.07	0.463	30.99	43.02	0.334
KDCoE ($i4$)	46.17	54.85	0.487	32.20	44.58	0.346
KDCoE (term)	**48.32**	**56.95**	**0.496**	**33.52**	**45.47**	**0.349**

Table 3.9: Results of cross-lingual entity alignment.

cess of KDCoE is terminated when either component is not able to propose ILLs for at least 1% of the entity vocabulary.

Results. Results are reported in Table 3.9, where the results by KDCoE are reported for three co-training iterations since the second iteration where KGEM is first leveraged, and for its final stage (which are respectively marked as KDCoE ($i2 - i4$) and KDCoE (term)). Among all baselines, MTransE-LT notably outperforms others, including other MTransE variants. The orthogonality constraint of OT seems to be too strict so that it impairs the performance. ITransE works well on aligning coherent monolingual KGs (Zhu et al., 2017), but does not adapt well to the inconsistent multilingual KGs. Without jointly adapting the monolingual vector spaces with the alignment, off-line approaches LM and CCA are left behind. On both language settings, KDCoE is able to gradually improve MTransE-LT in every iteration of co-training. The most significant improvements happen in the first iterations, where a majority of candidate ILLs are to be proposed. The final stages of KDCoE (6^{th} and 5^{th} iterations of the two settings) outperform the best baseline by almost doubling $Hit@1$ as well as offering significantly higher $Hit@10$ and MRR. Hence, the co-training approach of KDCoE on enhancing semi-supervised entity alignment is very promising.

Language	En-Fr			En-De		
Metric	$Hit@1$	$Hit@10$	MRR	$Hit@1$	$Hit@10$	MRR
Linear BOW	0.97	1.80	0.013	0.36	2.10	0.010
CNN	1.19	6.91	0.036	1.28	4.63	0.019
GRU	18.45	27.65	0.204	11.23	24.48	0.165
AGRU-mono	5.08	18.27	0.096	5.03	14.90	0.085
AGRU-multi	26.92	44.69	0.337	19.34	45.69	0.269
KDCoE ($i1$)	27.69	48.69	0.346	19.52	45.84	0.274
KDCoE ($i2$)	28.82	52.58	0.350	20.37	46.35	0.279
KDCoE ($i3$)	30.83	55.91	**0.384**	21.28	48.49	0.283
KDCoE (term)	**30.96**	**56.93**	0.382	**21.97**	**50.02**	**0.285**

Table 3.10: Results of zero-shot alignment.

Language	Fr				De			
Predict	Tail		Head		Tail		Head	
Metric	$Hit@10$	MRR	$Hit@10$	MRR	$Hit@10$	MRR	$Hit@10$	MRR
TransE	29.21	0.077	18.19	0.046	29.58	0.099	23.57	0.059
KDCoE-mono	31.05	0.092	16.88	0.053	29.13	0.124	27.63	0.106
KDCoE-cross	**37.21**	**0.139**	**22.23**	**0.093**	**34.17**	**0.134**	**31.05**	**0.143**

Table 3.11: Results of KG completion.

3.6.2 Zero-shot Alignment

This task focuses on aligning entities that do not exist in the structure of KG. While existing KG embedding models require candidates to occur for at least once in the KG structures, KDCoE is capable of dealing with zero-shot scenarios based on the representations of descriptions. For this task, we evaluate KDCoE by aligning the *zero-shot set* of WK3l60k, which are excluded from the KG structures for training. Meanwhile, we also compare the vanilla AGRU without co-training (AGRU-multi) against other encoding techniques, so as to show the effectiveness of our DEM. These baselines include the Linear BOW encoder that applies an affine layer to the averaged word embeddings of a description and the two-layer CNN with max-pooling in (Ji et al., 2017) that have been used to encode monolingual descriptions, as well as a two-layer GRU encoder without attention. We also substitute Bilbowa with monolingual Skipgram (Mikolov et al., 2013c) in AGRU (AGRU-mono) so as to verify the effectiveness of incorporating multilingual word embeddings.

Evaluation Protocol. We carry forward the corresponding configurations from the last experiment to show the performance under controlled variables. Specifically for CNN, we follow (Ji et al., 2017) to use 4-max-pooling and kernel-size of 2. Skipgram is trained separatedly on Wikipedia dumps of two languages towards 200-dimensional word vectors for AGRU-mono. All the baselines are trained on the ILL training set and corresponding descriptions. The results of KDCoE are reported for the first three iterations and the final stage.

Results. Results in Table 3.10 show that the vanilla DEM of AGRU outperforms the other encoders. This also indicates that AGRU is more competent for proposing ILLs in co-training based on unseen descriptions than others. As expected, co-training effectively leverages the zero-shot alignment with an increment of $Hit@1$ by 4.04% and 2.63%, as well as $Hit@10$ by 11.97% and 4.33% respectively on the two language settings. The results by GRU and AGRU-mono show that self-attention and multilingual word embeddings are vital to capture the cross-lingual semantic relatedness of descriptions from the word level. Failing to capture the sequence information, Linear BOW and CNN are left behind.

3.6.3 Cross-lingual KG Completion

Lastly, we compare the KGEM of KDCoE against its monolingual counterpart TransE for KG completion, based on the sparser French and German versions of WK3l60k. We explore with two prediction methods for KDCoE. *Monolingual prediction* (KDCoE-mono) aims to query the missing h or t of a triple (h, r, t) in the same way of TransE by searching among the entities of the same language to minimize the dissimilarity function $f_r(h, t)$ (Section 3.3.2). *Cross-lingual prediction* (KDCoE-cross) provides a new method of triple completion, by converting the monolingual prediction process to the embedding space of another language, then convert the results back to the source language. The idea of cross-lingual prediction is to leverage the traditional monolingual KG completion using a well-populated KG structure of an intermediary language given limited cross-lingual alignment.

Evaluation Protocol. We hold-out 10k French and German triples as test data. KDCoE is co-trained on the rest of the training data till termination. Cross-lingual predictions are processed in the space of English. TransE follows the configuration of KGEM in the previous experiments, and is trained on the KG structure of each language excluding the test data.

Results. The results for $Hit@10$ and MRR are reported in Table 3.11. KDCoE-mono performs at least comparably to TransE, which indicates that KDCoE preserves well the characterization of monolingual KG structures. Meanwhile, results of cross-lingual prediction prove feasibility of this new method by offering noticeably better outcomes than monolingual prediction. Although this experiment is relatively simple, and may subject to the adequacy of knowledge in the intermediary language, this method opens up a new direction of future work for this task. Moreover, suppose more languages of KGs are provided, we are interested in exploring an ensemble approach (Chen and Guestrin, 2016) that interpolates multiple KDCoEs on different bridges of languages to co-populate one sparse language-specific version of KG.

3.7 Conclusion

This chapter introduces the first work that generalizes knowledge graph embeddings to the multilingual scenario. Our model MTransE characterizes monolingual relations and compares three different techniques to learn cross-lingual alignment for entities and relations. Extensive experiments on the tasks of cross-lingual entity matching and triple alignment verification show that the linear-transformation-technique is the best among the three. Moreover, MTransE preserves the key properties of monolingual knowledge graph embeddings on monolingual tasks. Moreover, we propose a semi-supervised learning approach to co-train multilingual KG embeddings and the embeddings of entity descriptions for cross-lingual knowledge alignment. Our approach KDCoE effectively leverages KG embeddings for learning cross-lingual inferences on large, weakly-aligned KGs, which significantly outperforms previous models on the entity alignment task. The zero-shot alignment task also shows the effectiveness of KDCoE for improving the cross-lingual matching of entity descriptions through co-training. Meanwhile, we observe that KDCoE is able to enhance the traditional methods of KG completion by leveraging the information from another language.

CHAPTER 4

Transfer Embeddings with Complex Alignment Information

In this chapter, We extend the vanilla learning framework in the previous chapter to capture knowledge transfer with more complex alignment information. We consider the embedding learning of two-view knowledge bases, and biological knowledge graphs with fuzzy alignment.

4.1 Introduction

4.1.1 Ontology-level Concepts and Instance-level Entities

Several Knowledge bases, such as DBpedia (Lehmann et al., 2015), YAGO (Mahdisoltani et al., 2015) and ConceptNet (Speer et al., 2017), have incorporated knowledge graphs that can be categorized as two views: (i) the **instance-view knowledge graphs** that contain **relations** between specific **entities** in triples (for example, "*Barack Obama*", "*isPoliticianOf*", "*United States*") and (ii) the **ontology-view knowledge graphs** that constitute semantic **meta-relations** of abstract **concepts** (such as "*polication*", "*is leader of*", "*city*"). In addition, knowledge bases also provide **cross-view** links that connect ontological concepts and instances, denoting whether an instance is an instantiation from a specific concept.

Existing embedding models, however, are limited to only one single view, either on the instance-view graph (Bordes et al., 2013; Nickel et al., 2016; Yang et al., 2015b) or on the ontology-view graph (Chen et al., 2018d; Ristoski et al., 2018). Learning to represent a knowledge base from both views will no doubt provide more comprehensive insights. On one hand, instance embeddings provide detailed and rich information for their corresponding ontological concepts. For example, by observing many individual musicians, the embedding of its corresponding concept "*Musician*" can be largely determined. On the other hand, a concept embedding provides a high-level summary of its instances, which is extremely helpful when very few relations are observed for an instance. For example, for a musician who has very few relational facts in the instance-view graph, we can still tell his or her rough position in instance embedding space because he or she should not be far away from other musicians.

In this chapter, we first propose JOIE(Hao et al., 2019) to jointly embed the instance-view graph and the ontology-view graph, by leveraging (i) triples in both graphs and (ii) type links that connect the two graphs. It is a non-trivial task to effectively combine representation learning techniques on both views of a knowledge base together, which faces the following challenges: (i) the vocabularies of entities and concepts, as well as relations and meta-relations, are disjoint but semantically related in these two views of the knowledge base. The semantic mappings from entities to concepts and from relations to meta-relations are complicated and difficult to be precisely captured by any current embedding models; and (ii) the known type links often inadequately cover a vast number of entities, which leads to insufficient information to align both views of the knowledge base, and entails discovering new type links; (iii) the scales and topological structures are also largely inconsistent in the two views. Specifically, The ontological views are often sparser. They provide fewer types of relations and often form hierarchical substructures. In contrast, the instance view is much larger and heterogeneous in relation types.

To address the above issues, we propose a novel knowledge graph embedding model named `JOIE`, which jointly encodes both the ontology and instance views of a knowledge base. `JOIE` extends `MTransE` to support the representation learning. First, an *alignment model* associates the instance embedding to its corresponding concept embedding. Second, the *knowledge model* characterizes the relational facts of ontology and instance views in two separate embedding spaces, for which we also investigate several triple encoding techniques, as well as hierarchical aware en-

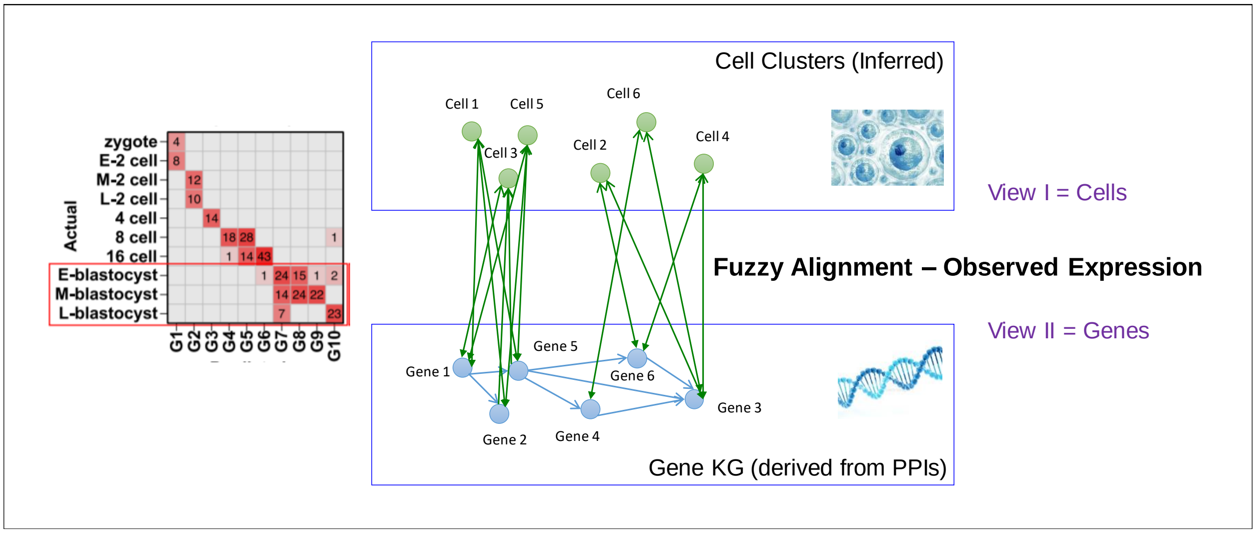


Figure 4.1: The scRNA-seq fuzzy alignment between genes and cells.

coding techniques for the ontology view. For the alignment model, we explore two techniques to capture the type links. The *cross-view grouping* technique assumes that the two views can be forced into the same embedding space, while the *cross-view transformation* technique enables non-linear transformations from the instance embedding space to the ontology embedding space. As for the knowledge embedding model, in particular, we use three state-of-the-art translational or similarity-based relational embedding techniques to capture the multi-relational structures of each view. Additionally, for some knowledge bases where ontologies constitute hierarchical substructures, we deploy a *hierarchy-aware* embedding technique based on knowledge non-linear transformations. This technique seek to help preserve the hierarchical property of such ontologies. Accordingly, we investigate with nine variants of JOIE and evaluate these models on two tasks: the triple completion task and the entity typing task. Experimental results on the triple completion task confirm the effectiveness of JOIE for populating knowledge in both ontology and instance-view knowledge graphs, and has significantly outperformed various baseline models. The results on the entity typing task show that our model is competent in discovering type links to align the ontology-view and the instance-view knowledge graphs.

4.1.2 Single-cell RNA-sequence Data As Fuzzy Alignment

Single-cell RNA-sequencing (scRNA-seq) enables high throughput measurement of RNA expression in individual cells, and seeks to help cell type identification and clustering (Gong et al., 2018; Li and Li, 2018; Talwar et al., 2018). The relations of genes (RNA) can be derived from the protein-protein interaction knowledge graphs (Szklarczyk et al., 2016), Hence, we further extend `MTransE` to deal with the single-cell RNA-sequencing. The proposed `KG-Transfer` model seeks to transfer the gene-level knowledge to the cell view potentially helps the inferences of cell information. Figure 4.1 shows an overview of the knowledge `KG-Transfer` seeks to represent.

Since the vocabularies of genes and cells are of largely different sizes, `KG-Transfer` distribute genes and cells in embedding spaces with different dimensionalities. The knowledge model encodes the relations of genes based on protein-protein interaction data. The cells does not have multi-relational data, while our objective here is to infer the cell clustering.

The additional challenge under this case lies mainly under the fuzzy alignment between genes and cells. As the scRNA-seq data measures different gene expressions between organisms, tissues, and disease states of a single cell based on wet lab transcripts. Due to that for each cell, the observed gene-cell associations typically have different number of observations and different evidential confidence in wet lab transcripts (Talwar et al., 2018). Hence, the alignment model needs to be adapted to capture such fuzzy alignment information. More over, due to technical limitations, scRNA-seq data often contain zero counts for many transcripts in individual cells (Wang et al., 2009). These zero counts, or dropout events, complicate the transfer learning by causing missing alignment information. Against this issue, we develop the alignment model of `KG-Transfer` as a Semi-Nonnegative Matrix Tri-factorization (Semi-NMTF) (Ding et al., 2010) based fuzzy alignment model between the gene and cell views. This alignment technique seeks to capture the associations between genes and cells based on fuzzy alignment and impute the missing values of the scRNA-seq matrix. We show that by transfering the gene-level knowledge, `KG-Transfer` is able to significantly improve cell clustering, especially under the case where the scRNA-seq data has extreme dropout rates and is highly sparse.

4.2 Joint Embedding of Ontological Concepts and Instance-view Entities

In this section, we introduce our proposed model JOIE for jointly embedding entities and concepts. We start with the formalization of two-view knowledge bases.

4.2.1 Formalization of Two-view Knowledge Bases

In a knowledge base, we use $\mathcal{G}_I$ and $\mathcal{G}_O$ to denote the instance-view knowledge graph and ontology-view knowledge graph respectively. The instance-view knowledge graph is denoted as $\mathcal{G}_I$, which is formed with $\mathcal{E}$, the set of entities, and $\mathcal{R}_I$, the set of relations. The set of concepts and meta-relations in the ontology-view graph $\mathcal{G}_O$ are similarly denoted as $\mathcal{C}$ and $\mathcal{R}_O$ respectively. Note that $\mathcal{E}$ and $\mathcal{C}$ (or $\mathcal{R}_I$ and $\mathcal{R}_O$) are disjoint sets. $(h^{(I)}, r^{(I)}, t^{(I)}) \in \mathcal{G}_I$ and $(h^{(O)}, r^{(O)}, t^{(O)}) \in \mathcal{G}_O$ denote triples in the instance-view knowledge graph and the ontology-view knowledge graph respectively, such that $h^{(I)}, t^{(I)} \in \mathcal{E}, h^{(O)}, t^{(O)} \in \mathcal{C}$, $r^{(I)} \in \mathcal{R}_I$, and $r^{(O)} \in \mathcal{R}_O$. Specifically, for each view in the knowledge base, a dedicated low-dimensional space is assigned to embed nodes and edges. Boldfaced $\mathbf{h}^{(I)}, \mathbf{t}^{(I)}, \mathbf{r}^{(I)}$ represent the embedding vectors of head entity $h^{(I)}$, tail entity $t^{(I)}$ and relation $r^{(I)}$ in one instance-view triple, which is denoted similarly for head and tail concepts $\mathbf{h}^{(O)}, \mathbf{t}^{(O)}$ connected with the meta-relation $\mathbf{r}^{(O)}$ for the ontology-view graph.

Besides the notations for two views, $\mathcal{S}$ is used to denote the set of known type links in the knowledge base, which contains associations between instances and concepts such as *"type_of"*. We use $(e, c) \in \mathcal{S}$ to denote a link between $e \in \mathcal{E}$ and its corresponding concept $c \in \mathcal{C}$. For example, (e: Los Angeles International Airport, c: airport) denotes that "*Los Angeles International Airport*" is an instance of the concept "*airport*". Looking into the nature of the ontology view, we also have hierarchical substructures identified by *"subclass_of"* (or equivalent meta-relations). That is, we can observe concept pairs $(c_l, c_h) \in \mathcal{T}$ that indicates that a finer (more specific) concept belongs to a coarser (more general) concept. One aforementioned example is (c_l: singer, c_h: person).

Our model JOIE consists of two model components that learn embeddings from the two views: the alignment model enables the connection and information flow between the two views by cap-

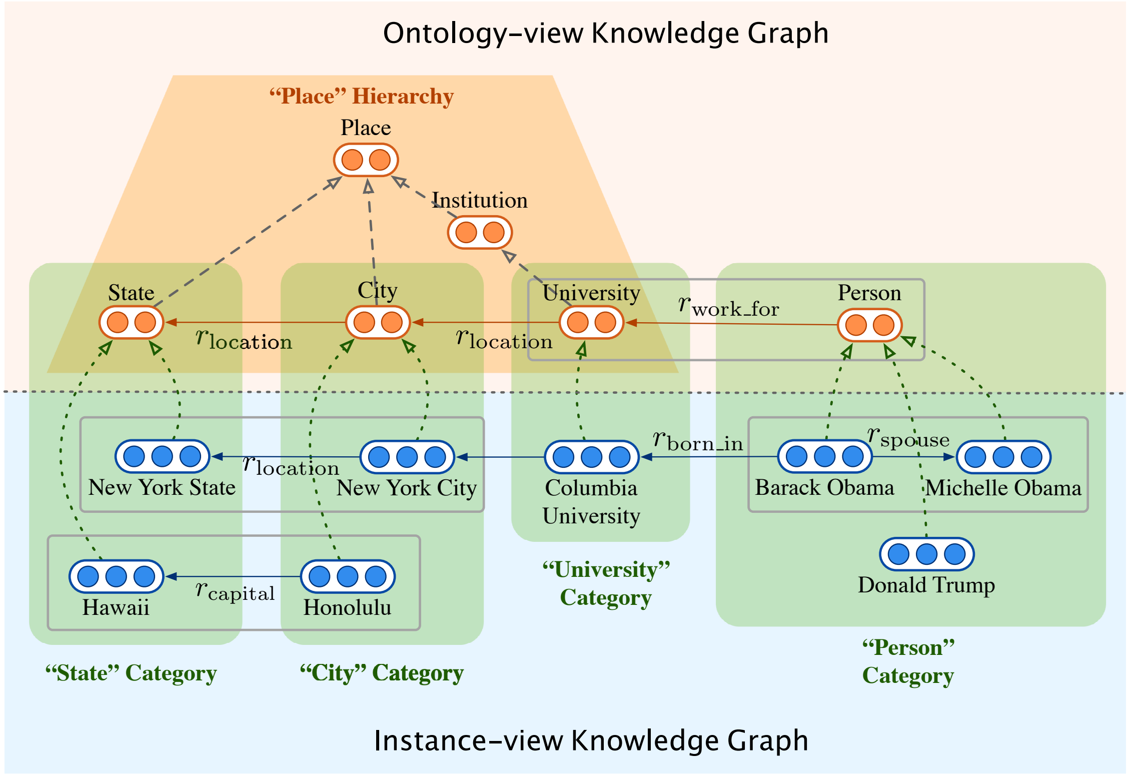
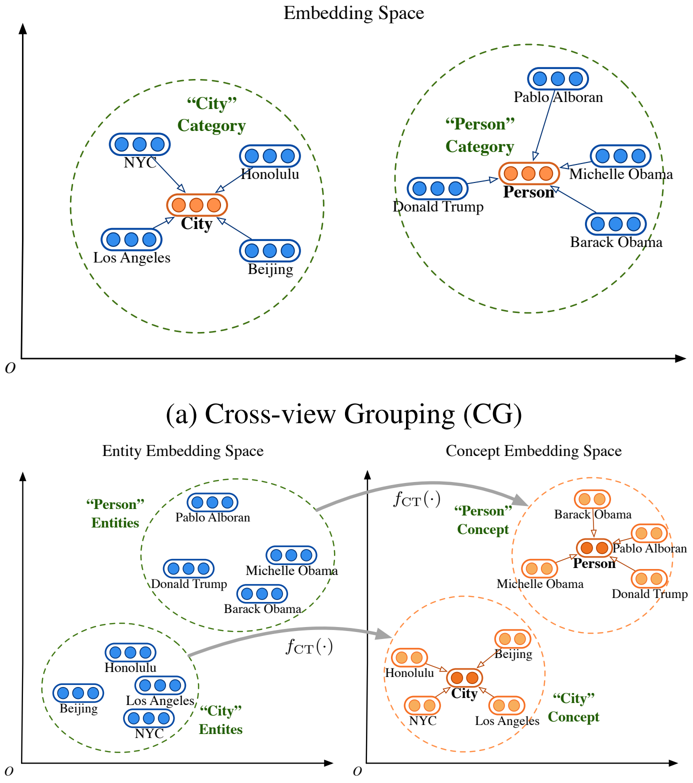


Figure 4.2: JOIE learns two aspects of a knowledge base. The alignment model learns embeddings from type links (dash arrows in green "category" box). The default knowledge model learns embeddings from triples (grey box) in each view; Besides, hierarchy-aware knowledge models the meta-relation facts that form hierarchies in the ontology (orange "Hierarchy" trapezoid).

turing the instantiation of entities from corresponding concepts, and the knowledge model encodes the entities/concepts and relations/meta-relations on each view of the knowledge base. The illustration of these model components for learning different aspects of the knowledge base is shown in Figure 4.2. In the following subsections, we first discuss the alignment model and knowledge model for each view, then combine them into variants of proposed JOIE model.

4.2.2 Alignment Model with Hierarchical Grouping Techniques

The goal of the alignment model is to capture the associations between the entity embedding space and the concept embedding space, based on the type links in knowledge bases, which will be our key contributions. We propose two techniques to model such associations: *Cross-view Grouping(CG)* and *Cross-view Transformation(CT)*. These two techniques are based on different assumptions and thus optimize different objective functions.



(a) Cross-view Grouping (CG)

(b) Cross-view Transformation (CT)

Figure 4.3: Intuition of the alignment model: Cross-view Grouping (a); Cross-view Transformation (b).

Cross-view Grouping (CG). The cross-view grouping method can be considered as grouping-based regularization, which assumes that the ontology-view knowledge graph and instance-view knowledge graph can be embedded into the same space, and forces any instance $e \in \mathcal{E}$ to be close to its corresponding concept $c \in \mathcal{C}$, as shown in Figure 4.3a. This requires the embedding dimensionalities for the instance-view and ontology-view graphs to be the same, i.e. $d = d_c = d_e$. Specifically, the categorical association loss for a given pair of cross-view link (e, c) is defined as the distance between the embeddings of e and c compared with margin γ^{CG}, and the loss is defined as,

$$J_{\text{A}}^{\text{CG}} = \frac{1}{|\mathcal{S}|} \sum_{(e,c)\in\mathcal{S}} \left[||\mathbf{c} - \mathbf{e}||_2 - \gamma^{\text{CG}}\right]_+, \tag{4.1}$$

where $[x]_+$ is the positive part of the input x, i.e. $[x]_+ = \max\{x, 0\}$. This penalizes the case where the embedding of e falls out the γ^{CG}-radius [1] neighborhood centered at the embedding of c. CG has a strong clustering effect that makes entity embeddings close to their concept embeddings in the end.

Cross-view Transformation (CT). We also propose a cross-view transformation technique, which seeks to transform information between the entity embedding space and the concept space. Unlike CG that requires the two views to be embedded into the same space, the CT technique allows the two embedding spaces to be completely different from each other, which will be aligned together via a transformation, as shown in Figure 4.3b. In other words, after the transformation, an instance will be mapped to an embedding in the ontology-view space, which should be close to the embedding of its corresponding concept:

$$\mathbf{c} \leftarrow f_{\text{CT}}(\mathbf{e}), \forall(e, c) \in \mathcal{S}, \tag{4.2}$$

where $f_{\text{CT}}(\mathbf{e}) = \sigma(\mathbf{W}_{\text{ct}} \cdot \mathbf{e} + \mathbf{b}_{\text{ct}})$ is a non-linear affine transformation. $\mathbf{W}_{\text{ct}} \in \mathbb{R}^{d_2 \times d_1}$ thereof is a weight matrix and $\mathbf{b}_{\text{ct}}$ is a bias vector. $\sigma(\cdot)$ is a non-linear activation function, for which we adopt $\tanh$.

[1] Typically, margin hyperparameter γ in the hinge loss can be chosen as 0.5 or 1 for different model settings. However, it is not a sensitive hyperparameter in our models.

Therefore, the total loss of the alignment model is formulated as Equation 4.3, which aggregates the CT objectives for all concepts involved in $\mathcal{S}$.

$$J_{\mathrm{A}}^{\mathrm{CT}} = \frac{1}{|\mathcal{S}|} \sum_{\substack{(e,c)\in\mathcal{S} \\ \wedge(e,c')\notin\mathcal{S}}} \left[\gamma^{\mathrm{CT}} + ||\mathbf{c} - f_{\mathrm{CT}}(\mathbf{e})||_2 - ||\mathbf{c}' - f_{\mathrm{CT}}(\mathbf{e})||_2\right]_+ \tag{4.3}$$

4.2.3 Knowledge Model

The aim of knowledge model is to preserve the original structural information in each view of the knowledge base separately in two embedding spaces. Because of the different semantic meanings of relations in the instance view and meta-relations in the ontology view, it helps to give each view separate treatment rather than combining them into a single representation schema, improving the performance of downstream tasks, as shown in Section 4.3.2. In this section, we provide two knowledge model techniques for encoding heterogeneous and hierarchical graph structures.

Default Knowledge Model To embed such a triple (h, r, t) in one knowledge graph, a score function $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$ measures the plausibility of it. A higher score indicates a more plausible triple. Any triple embedding technique is applicable in our knowledge framework. In this chapter, we adopt three representative techniques, i.e. translations (Bordes et al., 2013), multiplications (Yang et al., 2015b) and circular correlation (Nickel et al., 2016). The score functions of these techniques are given as follows.

$$\begin{aligned} f_{\mathrm{TransE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= -||\mathbf{h} + \mathbf{r} - \mathbf{t}||_2 \\ f_{\mathrm{Mult}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= (\mathbf{h} \circ \mathbf{t}) \cdot \mathbf{r} \\ f_{\mathrm{HolE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= (\mathbf{h} \star \mathbf{t}) \cdot \mathbf{r} \end{aligned} \tag{4.4}$$

where $\circ$ is the Hadamard product and $\cdot$ is the dot product. $\star : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ denotes circular correlation defined as $[\mathbf{a} \star \mathbf{b}]_k = \sum_{i=0}^{d} a_i b_{(k+i) \mod d}$.

To learn embeddings of all nodes in one graph $\mathcal{G}$, a hinge loss is minimized for all triples in the graph:

$$J_{\mathrm{K}}^{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{\substack{(h,r,t)\in\mathcal{G} \\ \wedge(h',r,t')\notin\mathcal{G}}} \left[\gamma^{\mathcal{G}} + f(\mathbf{h}', \mathbf{r}, \mathbf{t}') - f(\mathbf{h}, \mathbf{r}, \mathbf{t})\right]_+ , \tag{4.5}$$

where $\gamma_S > 0$ is a positive margin, and (h', r, t') is one sample from the set of corrupted triples which replace either head or tail entity and does not exist in $\mathcal{G}$.

The aforementioned techniques, losses and learning objectives for embedding graphs are naturally applicable for both instance-view graph and ontology-view graph. In the default knowledge model setting, for triples $(h^{(I)}, r^{(I)}, t^{(I)}) \in \mathcal{G}_I$ or $(h^{(O)}, r^{(O)}, t^{(O)}) \in \mathcal{G}_O$, we can compute $f_I(\mathbf{h}^{(I)}, \mathbf{r}^{(I)}, \mathbf{t}^{(I)})$ and $f_O(\mathbf{h}^{(O)}, \mathbf{r}^{(O)}, \mathbf{t}^{(O)})$ with the same techniques when optimizing $J_{\mathrm{K}}^{\mathcal{G}_I}$ and $J_{\mathrm{K}}^{\mathcal{G}_O}$. Combining the loss from instance-view and ontology-view graphs, the joint loss of the knowledge model is given as below,

$$J_{\mathrm{K}} = J_{\mathrm{K}}^{\mathcal{G}_I} + \alpha_1 \cdot\ J_{\mathrm{K}}^{\mathcal{G}_O}, \tag{4.6}$$

where a positive hyperparameter α_1 weighs between the structural loss of the instance-view graph and ontology-view graph.

In JOIE deployed with the default knowledge model, we employ the same triple encoding technique to represent both views of the knowledge base. The purpose of doing so is to enforce the same paradigm of characterizing relational inferences in both views. It is noteworthy that there are other triple encoding techniques for knowledge graph embeddings, which can potentially be used in our knowledge model. Since exploring different triple encoding techniques is not the focus of our chapter, we leave them as future work.

Hierarchy-Aware Knowledge Model for the Ontology It is observed that the ontology view of some knowledge bases form hierarchies, which is typically constituted by a meta-relation conforming the hierarchical property, such as "*subclass_of*" and "*is_a*" (Lehmann et al., 2015; Mahdisoltani et al., 2015). We can define such meta-relation facts as (c_l, r_{meta} = "*subclass_of*", c_h). For example, "*musician*" and "*singer*" belong to "*artist*" and "*artist*" is also subclass of "*person*". Such semantic ontological features requires additional modeling than other meta-relations. In other words, we further distinguish between meta-relations that form the ontology hierarchy and those regular semantic relations (such as *"related_to"*) in our knowledge model.

To address this problem, we propose the hierarchy-aware (HA) knowledge model by extending a similar method to that of cross-view transformation as defined in Equation 4.2. Given concept pairs (c_l, c_h), we model such hierarchies into a non-linear transformation between coarser concepts and associated finer concepts by

$$g_{\text{HA}}(\mathbf{c}_h) = \sigma(\mathbf{W}_{\text{HA}} \cdot \mathbf{c}_l + \mathbf{b}_{\text{HA}}) \tag{4.7}$$

where $\mathbf{W}_{\text{HA}} \in \mathbb{R}^{d_2 \times d_2}$ and $\mathbf{b}_{\text{HA}} \in \mathbb{R}^{d_2}$ are defined similarly. Also, we use $\tanh$ function as $\sigma(\cdot)$ option. This will introduce a new loss term, ontology hierarchy loss inside the ontology view, which is similar to Equation 4.3,

$$J_{\text{K}}^{\text{HA}} = \frac{1}{|\mathcal{T}|} \sum_{\substack{(c_l, c_h) \in \mathcal{T} \\ \wedge (c_l, c'_h) \notin \mathcal{T}}} \left[\gamma^{\text{HA}} + ||\mathbf{c}_h - g(\mathbf{c}_l)||_2 - ||\mathbf{c_h}' - g(\mathbf{c_l})||_2\right]_+ \tag{4.8}$$

Therefore, the total training loss of the hierarchy-aware knowledge model for both views changes slightly to,

$$J_{\text{K}} = J_{\text{K}}^{\mathcal{G}_I} + \alpha_1 \cdot J_{\text{K}}^{\mathcal{G}_O \setminus \mathcal{T}} + \alpha_2 \cdot J_{\text{K}}^{\text{HA}} \tag{4.9}$$

where positive α_1 and α_2 are two weighing hyperparameters. In Equation 4.9, $J_{\text{K}}^{\mathcal{G}_O \setminus \mathcal{T}}$ refers to the loss of the default knowledge model that is only trained on triples with regular semantic relations. J_{K}^{HA} is explicitly trained on the triples with meta-relations that form the ontology hierarchy, which is a major difference from Equation 4.6.

As the conclusion of this subsection, in `JOIE`, the basic assumption is that knowledge graphs have ontology hierarchy and rich semantic relational features compared to social or citation networks. `JOIE` is able to encode such knowledge graph properties in its model architecture. Note that we are also aware of the fact that there are more comprehensive properties of relations and meta-relations in the two views such as logical rules of relations and entity types. Incorporating such properties into the learning process is left as future work.

4.2.4 Joint Training on Two Views

Combining the knowledge model and alignment model, JOIE minimizes the following joint loss function:

$$J = J_{\mathrm{K}} + \omega \cdot J_{\mathrm{A}}, \tag{4.10}$$

where $\omega > 0$ is positive hyperparameter that balances between J_{K} and J_{A}.

Instead of directly updating J, our implementation optimizes $J_{\mathrm{K}}^{\mathcal{G}_I}$, $J_{\mathrm{K}}^{\mathcal{G}_O}$ and J_{A} alternately. In detail, we optimize $\theta^{\text{new}} \leftarrow \theta^{\text{old}} - \eta \nabla J_{\mathrm{K}}$ and $\theta^{\text{new}} \leftarrow \theta^{\text{old}} - (\omega\eta) \nabla J_{\mathrm{A}}$ in successive steps within one epoch. η is the learning rate, and ω differentiates between the learning rates for knowledge and cross-view losses.

We use the AMSGrad optimizer (Reddi et al., 2018) to optimize the joint loss function. We initialize vectors by drawing from a uniform distribution on the unit spherical surface, and initialize matrices using random orthogonal initialization (Saxe et al., 2014). During the training, we enforce the constraint that the L2 norm of all entity and concept vectors to be 1, in order to prevent them from shrinking to zero. This follows the setting by (Bordes et al., 2013; Nickel et al., 2016; Wang et al., 2014b; Yang et al., 2015b). Negative sampling is used on both knowledge model and alignment model with a ratio of 1 (number of negative samples per positive one). A hinge loss is applied for both models with all variants.

4.2.5 Variants of **JOIE** and Complexity

Without considering the HA technique, we have six variants of JOIE given two options of alignment models in Section 4.2.2 and three options of knowledge models in Section 4.2.3. For simplicity, we use the names of its components to denote specific variants of JOIE, such as "JOIE-TransE-CT" represents JOIE with the cross-view transformation and TransE-based default knowledge embeddings. In addition, we incorporate the hierarchy-aware knowledge model for the ontology view into cross-view transformation model[2], which produces three additional model variants

[2] We later show in the experiments that CT-based variants consistently outperform CG-based variants and thus we only apply HA knowledge model settings to CT-based model variants.

denoted as JOIE-HATransE-CT, JOIE-HAMult-CT, and JOIE-HAHolE-CT.

The model complexity depends on the alignment model and knowledge model for learning two-view knowledge bases. We denote n_e, n_c, n_r, n_m as the number of total entities, concepts, relations and meta-relations (typically $n_e \gg n_c$) and d_e, d_c as embedding dimensions ($d_e = d_c$ if CG is used). The model complexity of parameter sizes is $\mathcal{O}(n_e d_e + n_c d_c)$ for all CG-based variants and $\mathcal{O}(n_e d_e + n_c d_c + d_e d_c)$ for all CT-based variants. An additional parameter size of $\mathcal{O}(d_c^2)$ is needed if the hierarchy-aware knowledge model applies. Because of $n \gg d_e$ (or d_c), the parameter complexity is approximately proportional to the number of entities and the model training runtime complexity is proportional to the number of triples in the knowledge graph. For the task of triple completion in the knowledge graph, the time complexity for all variants is $\mathcal{O}(n_e d_e)$ for the instance-view graph or $\mathcal{O}(n_c d_c)$ for the ontology-view graph. To process each prediction case in the entity typing task, the time complexity is $\mathcal{O}(n_c d_e)$ for CG and $\mathcal{O}(n_c d_c d_e)$ for CT. Details about each task are curated in Section 4.3.2 and 4.3.3.

4.3 Entity Typing and Triple Completion

In this section, we evaluate JOIE with two groups of tasks: the triple completion task (Section 4.3.2) on both instance-view and ontology-view KGs and the entity typing task (Section 4.3.3) to bridge two views of the knowledge base. Besides, we provide a case study in Section 4.3.4 on ontology population and long-tail entity typing. We also present hyperparameter study, effects of cross-view sufficiency and negative samples.

4.3.1 Datasets

To the best of our knowledge, existing datasets for knowledge graph embeddings consider only an instance view (e.g. FB15k (Bordes et al., 2013)) or an ontology view (e.g. WN18 (Bordes et al., 2014a)). Hence, we prepare two new datasets: *YAGO26K-906* and *DB111K-174*, which are extracted from YAGO (Mahdisoltani et al., 2015) and DBpedia (Lehmann et al., 2015) respectively.

Table 4.1 provides the statistics of both datasets. Normally, the instance-view knowledge graph is significantly larger than the ontology-view graph. Also, we notice that the two knowledge

Table 4.1: Statistics of datasets.

Dataset	Instance Graph $\mathcal{G}_I$			Ontology Graph $\mathcal{G}_O$			Type Links $\mathcal{S}$
	#Entities	#Relations	#Triples	#Concepts	#Meta-relations	#Triples	
YAGO26K-906	26,078	34	390,738	906	30	8,962	9,962
DB111K-174	111,762	305	863,643	174	20	763	99,748

bases are different in the density of type links, i.e., DB111K-174 has a much higher entity-to-concept ratio (around 643.4) than YAGO26K-906 (around 28.7). Both datasets are available at `Anonymous_URL` (Link, 2019).

4.3.2 Triple Completion

The objective of triple completion is to construct the missing relation facts in a knowledge graph structure, which directly tests the quality of learned embeddings. In our experiment, this task spans into two sub-tasks for instance-view knowledge graph completion and ontology population. We perform the sub-tasks on both datasets with all JOIE variants compared with baseline models.

Evaluation Protocol First, we separate the instance-view triples into training set $\mathcal{G}_I^{\text{train}}$, validation set $\mathcal{G}_I^{\text{valid}}$ and test set $\mathcal{G}_I^{\text{test}}$, as well as separate similarly the ontology-view triples to $\mathcal{G}_O^{\text{train}}$, $\mathcal{G}_O^{\text{valid}}$ and $\mathcal{G}_O^{\text{test}}$. The percentage of the training, validation and test cases is approximately 85%, 5% and 10%, which is consistent to that of the widely used benchmark dataset (Bordes et al., 2013) for instance-only knowledge graph embeddings. Each JOIE variant is trained on $\mathcal{G}_I^{\text{train}}$ and $\mathcal{G}_O^{\text{train}}$ triples along with all type links $\mathcal{S}$. In the testing phase, given each query $(h, r, ?t)$, the plausibility scores $f(\mathbf{h}, \mathbf{r}, \tilde{\mathbf{t}})$ for triples formed with every $\tilde{t}$ in the test candidate set are computed and ranked by the knowledge model. We report three metrics for testing: mean reciprocal ranks (MRR), accuracy ($Hits$@1) and the proportion of correct answers ranked within the top 10 ($Hits$@10). All three metrics are preferred to be higher, so as to indicate better triple completion performance. Also, we adopt the filtered metrics as suggested in previous work which are aggregated based on the premise that the candidate space has excluded the triples that have been seen in the training set (Bordes et al., 2013; Yang et al., 2015b).

As for the hyperparameters in training, we select the dimensionality d among $\{50, 100, 200, 300\}$ for concepts and entities, learning rate among $\{0.0005, 0.001, 0.01\}$, margin γ among $\{0.5, 1\}$.

We also use different batch sizes according to the sizes of graphs. We fix the best configuration $d_e = 300, d_c = 50$ for CT and $d_e = d_c = 200$ for CG with $\alpha_1 = 2.5, \alpha_2 = 1.0$. We set $\gamma^{\mathcal{G}_I} = \gamma^{\mathcal{G}_O} = 0.5$ as the default for all TransE variants and $\gamma^{\mathcal{G}_I} = \gamma^{\mathcal{G}_O} = 1$ for all Mult and HolE variants. The training processes on all datasets and models are limited to 120 epochs.

Baselines We compare our model with TransE, DistMult and HolE as well as TransC (Lv et al., 2018). We deploy the following variants of baselines: (i) We train these mono-graph models (TransE, DistMult and HolE) either on instance-view triples or ontology-view triples separately, denoted as (*base*) in Table 4.2; (ii) We also train TransE, DistMult and HolE based on all triples in both $\mathcal{G}_I^{\text{train}}$ and $\mathcal{G}_O^{\text{train}}$. For the second setting thereof, we incorporate type links by adding one additional relation "*type_of*" to them, denoted as (*all*) in Table 4.2. (iii) TransC is trained on both views of a knowledge base. TransC is a recent work that differentiates between the encoding process of concepts from instances. Note that TransC is equivalent to a simplified case of our JOIE-TransE-CG where no semantic meta relations in the ontology view are included. For that reason, TransC does not apply to the completion of the ontology view.

Results As reported in Table 4.2, we categorize the results into three different groups based on the knowledge models. Though three knowledge models have different capabilities, among all the baselines in same group, JOIE notably outperforms others by 6.8% on MRR, and 14.8% on $Hit@10$ on average. A significant improvement is achieved on the ontology-view of DB111K-174 with JOIE compared to concept embeddings trained with only ontology-view triples and even 10.4% average increment compared to "all"-setting baselines and 34.97% compared to "base"-setting baselines. These results indicate that JOIE has better ability to utilize information from the instance view to promote the triple completion in ontology view. Comparing different knowledge models, translation based models performs better than similarity based models on ontology population and instance-view knowledge graph completion on the DB111K-174dataset. This is because these graphs are sparse, and TransE is less hampered by the sparsity in comparison to the similarity-based techniques (Pujara et al., 2017). By applying the HA technique in the knowledge models with CT, the performance on instance-view triple completion is noticeably improved in most cases in comparison to the default knowledge CT-based models, especially in variants with

Table 4.2: Results of triple completion. Note that H@1 and H@10 denote $Hit@1$ and $Hit@10$ respectively. For each group of model variants with the same knowledge encoding techniques, the best results are bold-faced. The overall best results on each dataset are under-scored.

Datasets	YAGO26K-906						DB111K-174					
Graphs	$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion			$\mathcal{G}_I$ KG Completion			$\mathcal{G}_O$ KG Completion		
Metrics	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10
TransE (base)	0.195	14.09	34.51	0.145	12.29	20.59	0.327	22.26	49.01	0.313	23.22	46.91
TransE (all)	0.187	13.73	35.05	0.189	14.72	24.36	0.318	22.70	48.12	0.539	47.90	61.84
TransC	0.252	15.71	37.79	–	–	–	0.359	24.83	49.31	–	–	–
JOIE-TransE-CG	0.264	16.38	35.45	0.189	11.16	29.44	0.394	27.75	51.20	0.598	53.84	71.79
JOIE-TransE-CT	0.292	**18.72**	44.14	0.240	14.49	33.47	0.443	32.10	67.89	**0.622**	**58.10**	72.97
JOIE-HATransE-CT	**0.306**	18.62	**51.72**	**0.263**	**16.72**	**38.46**	**0.473**	**33.79**	**71.37**	0.591	52.07	**79.65**
DistMult (base)	0.253	22.91	28.76	0.197	**17.72**	25.08	0.265	25.95	27.63	0.235	15.18	29.11
DistMult (all)	0.288	**24.06**	31.24	0.156	14.32	16.54	0.280	27.24	29.70	0.501	45.52	64.73
JOIE-Mult-CG	0.274	18.80	37.45	0.198	11.16	27.91	0.320	23.44	49.49	0.532	46.15	68.91
JOIE-Mult-CT	**0.309**	20.40	**46.15**	**0.207**	14.71	30.43	**0.404**	**26.55**	**60.86**	**0.563**	**50.50**	71.62
JOIE-HAMult-CT	0.296	19.39	45.48	0.202	13.72	**31.10**	0.369	24.82	55.86	0.521	38.46	**77.25**
HolE (base)	0.265	**25.90**	28.31	0.192	18.70	20.29	0.301	29.24	31.51	0.227	18.91	32.83
HolE (all)	0.252	24.22	26.56	0.138	11.29	14.43	0.295	28.70	30.32	0.432	38.80	56.05
JOIE-HolE-CG	0.253	18.75	34.11	0.167	13.04	22.33	0.361	24.13	46.15	0.469	41.89	62.16
JOIE-HolE-CT	0.313	20.40	47.80	0.229	**20.85**	28.42	0.425	29.09	66.88	**0.514**	**43.24**	69.23
JOIE-HAHolE-CT	**0.327**	22.42	**52.41**	**0.236**	16.72	**30.96**	**0.464**	**33.11**	**69.56**	0.503	40.80	**71.03**

translation and circular correlation based knowledge models.

Generally, JOIE provides an effective method to train two-view knowledge base separately and both $\mathcal{G}_I$ and $\mathcal{G}_O$ benefit each other in learning better embeddings, producing promising results in the triple completion task.

4.3.3 Entity Typing

The entity typing task seeks to predict the associating concepts of certain given entities. Similar to the triple completion task, we rank all candidates and report the top-ranked answers for evaluation.

Evaluation Protocol We separate the type links of each dataset into training and test sets with the ratio of 60% to 40%, denoted as $\mathcal{S}^{\text{train}}$ and $\mathcal{S}^{\text{test}}$ respectively. Each model is trained on the entire instance-view and ontology-view graphs with type links $\mathcal{S}^{\text{train}}$. Hyperparameters are carried forward from the triple completion task, in order to evaluate under controlled variables. In the test phase, given a specific entity e_q, we rank the concepts based on their embedding distances from

Table 4.3: Results of entity typing.

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
TransE	0.144	7.32	35.26	0.503	43.67	60.78
MTransE	0.689	60.87	77.64	0.672	59.87	81.32
JOIE-TransE-CG	0.829	72.63	93.35	0.828	70.58	95.11
JOIE-TransE-CT	0.843	75.31	93.18	0.846	74.41	94.53
JOIE-HATransE-CT	**0.897**	**85.60**	**95.91**	**0.857**	**75.55**	**95.91**
DistMult	0.411	36.07	55.32	0.551	49.83	68.01
JOIE-Mult-CG	0.762	62.62	87.82	0.764	60.83	91.80
JOIE-Mult-CT	0.805	70.83	89.25	**0.791**	65.30	**93.47**
JOIE-HAMult-CT	**0.865**	**81.63**	**91.83**	0.778	**69.38**	85.71
HolE	0.395	34.83	54.79	0.504	44.75	65.38
JOIE-HolE-CG	0.777	65.30	87.89	0.784	66.75	89.37
JOIE-HolE-CT	0.813	72.27	88.71	0.805	68.84	**91.22**
JOIE-HAHolE-CT	**0.888**	**83.67**	**93.87**	**0.808**	**72.51**	89.79

the projection of $\mathbf{e}_q$ in the concept embedding space. and calculate MRR, $Hit@1$ (i.e. accuracy) and $Hit@3$ on the test queries. We perform the entity typing task on both datasets with all JOIE variants compared with these baselines.

Baselines We compare with TransE, DistMult, HolE and MTransE. For baselines other than MTransE, we convert the type links (e, c) to triples (e, r_T=“*type_of*”, c). Therefore, entity typing is equivalent to the triple completion task for these baseline models. For MTransE, we treat concepts and entities as different views (originally input as knowledge bases of two languages in (Chen et al., 2017b)) in their model and test with distance-based ranking.

Results Results are reported in Table 4.3. All JOIE variants perform significantly better than the baselines. The best JOIE model, i.e. JOIE-TransE-CT, outperforms the best baseline model MTransE by 15.4% in terms of accuracy and 14.4% in terms of MRR on YAGO26K-906. The improvement on accuracy and MRR are 14.3% and 14.5% on DB111K-174 compared to MTransE. The results by other baselines confirm that the type links, which apply to all entities and concepts, cannot be properly captured as a regular relation and requires a dedicated representation technique.

Considering different JOIE variants, our observation is that using translation based knowledge model and CT as the alignment model (JOIE-TransE-CT) is consistently better than other settings on both datasets. It has an average of 4.1% performance gain in MRR over JOIE-HolE-CT and JOIE-DistMult-CT, and an average of 2.17% performance gain in accuracy over the best of the rest variants (JOIE-TransE-CG). We believe that, compared with similarity-based knowledge models, translation based knowledge model better differentiates between different entities and different concepts in knowledge graphs with directed relations and meta-relations in the knowledge base (Pujara et al., 2017). The results by CT-based model variants are generally better than those by CG-based ones. We believe this is due to two reasons: (i) CT allows the two embedding spaces have different dimensionalties, and hence better characterizes the ontology-view that is smaller and sparser than the instance view; (ii) As the topological structures of the two views may exhibit some inconsistency, CT adapts well and is less sensitive to such inconsistency than CG.

In terms of different knowledge models, it is also observed that HA knowledge model with CT settings can drastically enhance entity typing task and achieve the best performance especially for YAGO26K-906 with relatively rich ontology, which improves an average of 6.0% on MRR and 10.5% in accuracy compared with the default knowledge settings. The reason that the HA technique does not have similar effects on DB111K-174 is because DB111K-174 contains a small ontology with much smaller hierarchical structures[3]. Comparing the two datasets, our experiments show that, JOIE generally achieves similar accuracies and MRR scores on YAGO26K-906 and DB111K-174, but slightly better $Hit@3$ on DB111K-174 due to its smaller candidate space.

Our method opens up a new direction that the learned embedding may help guide labeling entities with unknown types. In Section 4.3.4 and Appendix 4.4, we provide more experiments and insights on the benefits of representation learning with JOIE.

4.3.4 Case Study

In this section, we provide two case studies for ontology population and entity typing for long-tail entities.

Ontology Population By embedding the meta-relations and concepts in the ontology view, the

[3] DB111K-174 contains 164 ontology-view triples for meta-relations with the hierarchical property, while YAGO26K-906 contains 1,411.

triple completion process can already populate the ontology view with seen meta-relations, by answering the query like (“*Concert*”,“*Related to*”,$?t$) in the knowledge graph completion task. Given the top answers of the query, we can reconstruct triples like (“*Concert*”,“*Related to*”,“*Ballet*”) and (“*Concert*”,“*Related* to”,“*Musical*”) with high confidence. However, this process does not resolve the zero-shot cases where some concepts may satisfy some meta-relations that have not pre-existed in the vocabulary of meta-relations. We cannot predict the potentially new meta-relation "is Politician of" directly with triple completion by answering the following query: (“*Office Holder*”, $?r$, “*Country*”).

Our proposed `JOIE` provides a feasible solution by leveraging the alignment model that bridges the two views of the knowledge graph, and migrate proper instance-view relations to ontology-view meta-relations. This is realized by transforming the concept embeddings in the query to the entity embedding space, and selecting candidate relations from the instance-view. Considering the previous query (“*Office Holder*”, $?r$, “*Country*”), we first find the concept embeddings of “*Office Holder*” and “*Country*” (denoted as $\mathbf{c}_{\text{office}}$ and $\mathbf{c}_{\text{country}}$ respectively), and then transform them to the entity space. Specifically, for `JOIE` variants with translational knowledge model, we find the instance-view relations that are closest to $f_{\text{CT}}^{\text{inv}}(\mathbf{c}_{\text{country}}) - f_{\text{CT}}^{\text{inv}}(\mathbf{c}_{\text{office}})$. Figure 4.4 shows the PCA projections of the top 10 relation prediction results for this query. The top 3 relations are “*is Politician of*”, “*is Leader of*” and “*is Citizen of*”, which are all reasonable answers.

Table 4.4 shows some examples of newly discovered meta-relation facts that have not pre-existed in the ontology views of the two datasets. Five predictions with the highest plausibility (smallest distance) are provided for each query from the ontology-view graph[4]. From these top predictions, we observe that most populated ontology triples migrated from the instance view are meaningful.

Long-tail entity typing In knowledge graphs, the frequency of entities and relations often follow a long-tail distribution (Zipf’s law). In this case study, we select the entities with considerably low frequency[5], which involve around 15%-30% of total entities in the instance view of the two

[4] The first two queries are from YAGO26K-906 and the rest are from DB111K-174.

[5] In this experiment, we select entities in YAGO26K-906 which occurs less than 8 times (15% least frequent entities)

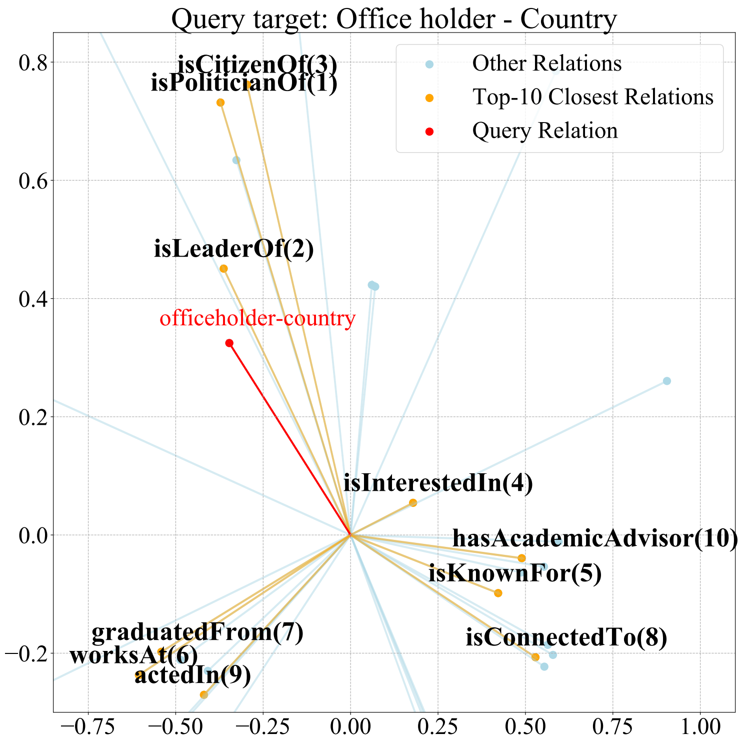


Figure 4.4: Examples of ontology population by finding the closest relations in the instance view for the query "Office Holder-Country". Top 10 predicted relations are plotted with their ranks.

knowledge base datasets. Then, we evaluate the entity typing task for these long-tail entities. Table 4.5 shows the results by the best baselines (DistMult, MTransE) and a groups of our best JOIE variants. Similar to our previous observation, JOIE significantly outperforms other baselines. Compared with the results in Section 4.3.3, we observe the depletion of performance for all models, while JOIE variants only have an average of 12.5% decrease in MRR with CG models and 12.3% decrease in MRR with CT models while other baselines suffer over 20% on long-tail entity prediction. There is also an interesting observation that, for long-tails entities, smaller embeddings for both CG ($d_1 = d_2 = 100$) and CT ($d_1 = 100, d_2 = 50$) models are beneficial for associated concept prediction. We hypothesize that this is caused by overfitting on long-tail entities if high dimensionality is used for training without enough training data.

In Table 4.6, we include some examples of top 3 predicted categories of long-tail entities by DistMult, MTransE and JOIE (using JOIE-HATransE-CT variant) from DB111K-174, when the

and entities in DB111K-174 which occurs less than 3 times (15% least frequent entities).

Table 4.4: Examples of ontology population from JOIE-TransE-CT. Top 5 Populated Triples with smallest L2-norm distances are provided with reasonable answers bold-faced.

Query	Top 5 Populated Triples with distances
(scientist,?r, university)	scientist, ***graduated from***, university (0.499)
	scientist, ***isLeaderOf***, university (1.082)
	scientist, *isKnownFor*, university (1.098)
	scientist, *created*, university (1.119)
	scientist, ***livesIn***, university (1.141)
(boxer, ?r, club)	boxer, ***playsFor*** , club (1.467)
	boxer, ***isAffiliatedTo*** , club (1.474)
	boxer, ***worksAt*** , club (1.479)
	boxer, *graduatedFrom* , club (1.497)
	boxer, *isConnectedTo* , club (1.552)
(TV station, ?r, country)	TV station, ***headquarter***, country (1.221)
	TV station, *parentOrganisation*, country (1.246)
	TV station, *appointer*, country (1.253)
	TV station, ***broadcastArea***, country (1.266)
	TV station, ***principalArea***, country (1.271)
(scientist, ?r, scientist)	scientist, *deputy*, scientist (0.204)
	scientist,***doctoralAdvisor***, scientist (0.218)
	scientist, ***doctoralStudent***, scientist (0.221)
	scientist, ***relative***, scientist (0.228)
	scientist, ***spouse***, scientist (0.230)

instance-view graph and ontology-view graph are relatively sparser. JOIE is still able to make correct predictions of low-frequency entities while other baselines models can only output inaccurate predictions.

4.4 Ablation Study

In this section, we provide some insights on several critical factors that affect the performance of the model. These include the embedding dimensionality, sufficiency of type links in training, and

Table 4.5: Results of long-tail entities typing.

Datasets	YAGO26K-906			DB111K-174		
Metrics	MRR	Acc.	Hit@3	MRR	Acc.	Hit@3
DistMult	0.156	10.89	25.33	0.219	16.48	33.71
MTransE	0.526	46.45	67.25	0.505	46.67	64.36
JOIE-TransE-CG	0.708	59.97	79.80	0.741	64.45	83.05
JOIE-TransE-CT	0.737	62.05	82.60	0.758	66.35	83.80
JOIE-HATransE-CT	**0.802**	**69.66**	**87.75**	**0.760**	**67.34**	**89.79**

Table 4.6: Examples of long-tail entity typing. Top 3 predictions are provided with the correct type bold-faced.

Entity	Model	Top 3 Concept Prediction
Laurence Fishburne	DistMult	football team, club, team
	MTransE	writer, **person**, artist
	JOIE	**person**, artist, philosopher
Warangal City	DistMult	country, village,**city**
	MTransE	administrative region, **city**, settlement
	JOIE	**city**, town, country
Royal Victor -ian Order	DistMult	person, writer, administrative region
	MTransE	election, award, **order**
	JOIE	award, **order**, election

the effect of adopting negative sampling in alignment models.

4.4.1 Dimensionality

Dimensionality is a key hyperparameter that affects the quality of the obtained embeddings. Figure 4.5a shows the MRR of model variants with the CG-based cross-view association according to different embedding dimensions d. It is observed in Figure 4.5a that the performance of CG variants are generally improving from $d = 50$ to $d = 200$, however, after reaching the optimal $d_{\text{opt}} = 200$, MRR begins to drop at $d = 300$. Similarly we plot MRR scores for both dataset with CT model variants in Figure 4.5b.

We compare four different dimensionality settings of (d_1, d_2): $(100, 20)$,$(100, 50)$,$(300, 50)$ and $(300, 100)$[6]. Most of the JOIE variants achieve their best performance under the embedding setting $(d_1, d_2) = (300, 50)$ rather than $(d_1, d_2) = (300, 100)$ (except JOIE-Mult-CT on DB111K-174). The reason is that, JOIE set with low dimensionalities easily falls short of capturing latent features of entities and concepts, while too high dimensionalities lead to overfitting on the ontology view of knowledge graph, as well as inefficient training and prediction processes.

[6] $(d_1, d_2) = (100, 20)$ denotes that entities are embedded with $d_1 = 100$ dimensional vectors and concepts are embedded with $d_2 = 20$ dimensional vectors

Table 4.7: Effects of negative sampling in type links

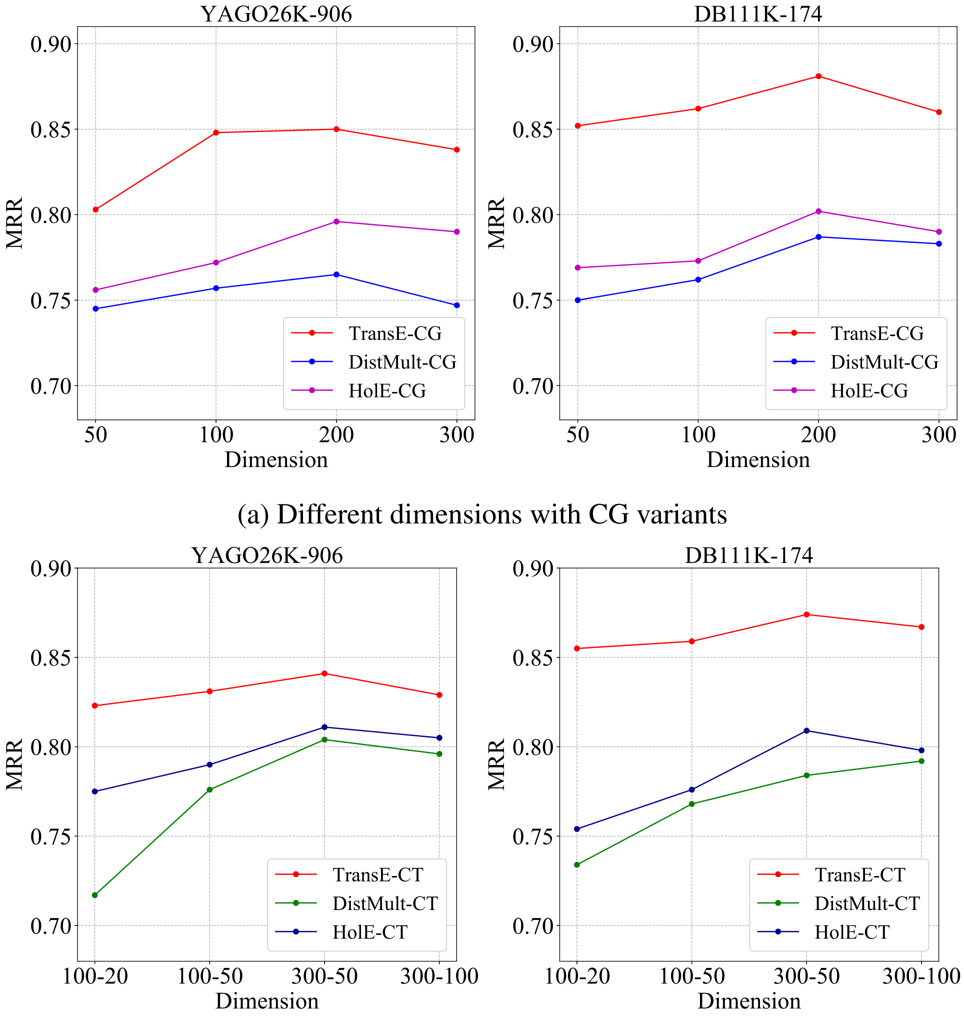
Datasets	YAGO26K-906		DB111K-174	
Setting	W/O NS	W/ NS	W/O NS	W/ NS
JOIE-TransE-CG	0.657	0.805	0.815	0.864
JOIE-Mult-CG	0.627	0.762	0.761	0.797
JOIE-HolE-CG	0.682	0.777	0.783	0.815
JOIE-TransE-CT	0.501	0.847	0.667	0.883
JOIE-Mult-CT	0.490	0.829	0.494	0.811
JOIE-HolE-CT	0.508	0.821	0.560	0.821

4.4.2 Sufficiency of Type Information

type links between the instance-view graph and the ontology-view graph are key components, which bridge and enable the information flow between two views to generate embeddings. We also investigate the influence of type links and their sufficiency in training.

We define the train set ratio $\nu = \{0.2, 0.4, 0.6, 0.8\}$, which means the proportions of the type links that are used for training JOIE. MRR score is reported in Figure 4.6a on YAGO26K-906 and Figure 4.6b on DB111K-174. As expected, when the proportion of type links used for training increasing from 20% to 80%, the performance improves by 3.2% on YAGO26K-906 and by 2.9% on DB111K-174 in terms of MRR. It is noteworthy that JOIE trained with 20% type links still outperforms MTransE trained with 60% type links, which indicates that one advantage of JOIE is its outstanding generalization ability to other untyped entities, given limited knowledge on entity-concept pairs.

One interesting observation is that, when ν increases from 0.6 to 0.8, the performance of CG variants does not necessarily improve, while the performance of CT variants still has significant improvements. We hypothesize that this is because the strong clustering-based constraint in CG can be sensitive to even minor inconsistencies between the topological structures of the two knowledge graph views, giving too much supervision. CT, on the contrary, is more robust against the inconsistency between the two views. There is a trade-off between the robustness of CT and the efficiency of CG.



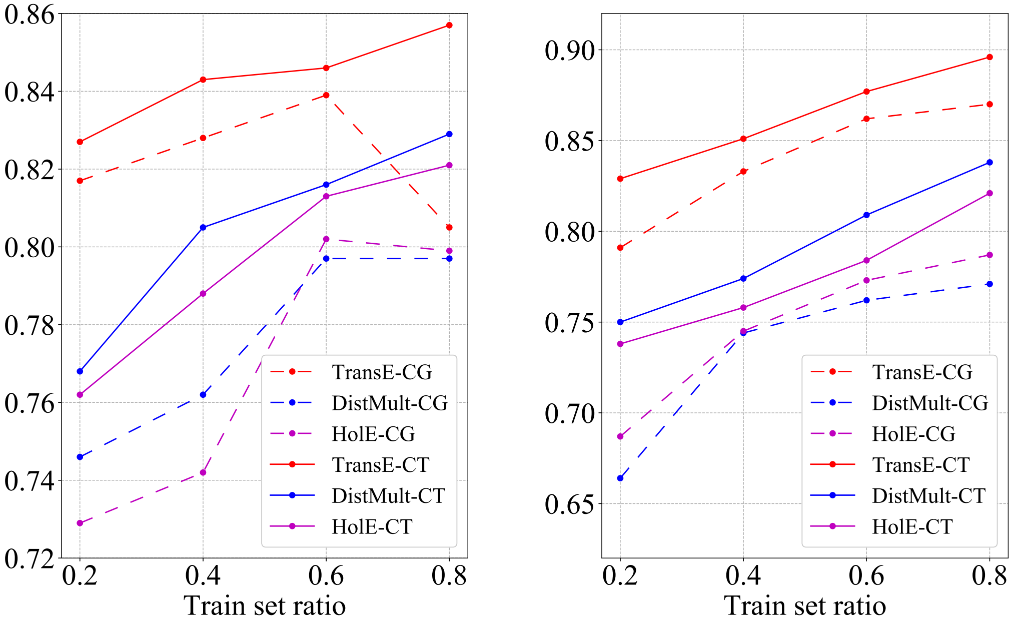
(a) Different dimensions with CG variants

(b) Different dimensions with CT variants

Figure 4.5: Performance of entity typing task on both datasets with different entity and concept embedding dimensionalities

4.4.3 Effects of Negative Sampling

Negative sampling is widely applied in the encoding process of a single knowledge graph structure (Bordes et al., 2013; Yang et al., 2015b). One interesting question is whether to use negative sampling for capturing the type links between two structures, i.e. to provide corrupted entity-concept pairs such as (“Barack Obama”,“state”). We compare the results of entity typing task by JOIE variants with and without cross-view link negative samples in Table 4.7. It is our finding that there is a significant performance drop if negative sampling is disabled in CT, while negative



(a) YAGO26K-906

(b) DB111K-174

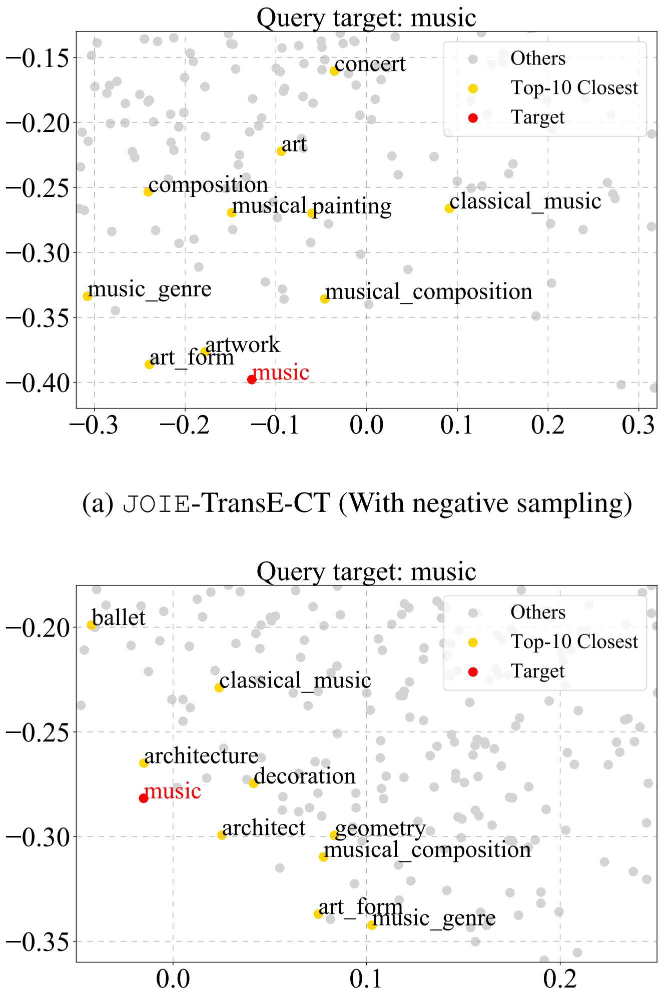
Figure 4.6: The effect of training the model using different proportions of type links on (a) YAGO26K-906 and (b) DB111K-174

sampling has less effect on CG. We hypothesize that the difference is attributed to the fact that strong clustering-based constraint of CG is already effective in separating irrelevant concepts.

We show the effects of negative sampling by visualizing the results of one query, which are plotted as PCA projections in Figure 4.7. For the displayed query which targets at the concept "music", we plot the 10 nearest neighbors of concepts. Although related concepts such as "classic music", "concert" and "artist movement" still stay close by "music" in both settings, other irrelevant concepts including "decoration" and "architect" intercept in JOIE-TransE-CT without negative sampling. We find such phenomenon frequently exist in the JOIE embeddings trained without negative sampling, which no-doubt impairs the performance of the entity typing task.

4.5 Transfer Gene Knowledge Based on Fuzzy Alignment

In this section, we switch gear to introduce KG-Transfer. Under this circumstance, we carry forward the notations in Section 4.2. We use $\mathcal{E}$ and $\mathcal{C}$ to denote the sets of genes and cells respec-



(a) JOIE-TransE-CT (With negative sampling)

(b) JOIE-TransE-CT (Without negative sampling)

Figure 4.7: Visualizing the effects on embeddings of negative sampling on type links

tively. $\mathcal{E} \subset \mathbb{R}^{|\mathcal{E}|\times d_e}$ and $\mathcal{E} \subset \mathbb{R}^{|\mathcal{C}|\times d_c}$ denote the embedding matrices of $\mathcal{E}$ and $\mathcal{C}$, for which d_e and d_c are again the dimensionalities. $\mathcal{X} \subset \mathbb{Z}_{\geq}^{|\mathcal{E}|\times|\mathcal{C}|}$ is a non-negative integer matrix that denotes the scRNA-seq data. Each entry $\mathcal{X}_{ij}$ corresponds to the transcript count for a pair of gene e_i and cell c_j.

KG-Transfer introduces a Semi-NMTF-based alignment model, which minimizes the following constrained loss function,

$$J_{\mathrm{A}}^{\mathrm{SNMTF}} = \left\| \mathcal{X} - \mathcal{E}\mathcal{S}\mathcal{C}^{\top} \right\|_F \text{ s.t. } \mathcal{E} \geq 0 \text{ and} \mathcal{C} \quad \geq 0 \tag{4.11}$$

in which $\mathcal{S} \subset \mathbb{R}^{d_e \times d_c}$ is an intermediate hidden matrix. The joint loss function of the learning objective (i.e. previously Equation 4.10) is rewritten as follows:

$$J = J_{\text{K}}^{\mathcal{G}_I} + \omega_1 \cdot J_{\text{A}}^{\text{SNMTF}} + \omega_2 \left\|\mathcal{E}\right\| + \omega_2 \left\|\mathcal{C}\right\| \tag{4.12}$$

In Equation 4.12, the knowledge model does not train on a $\mathcal{G}_O$, since the cell view does not form a knowledge graph for training. The to make the embeddings tractable for optimization by the Semi-NMTF based alignment model, instead of enforcing an embedding normalization constraint on $\mathcal{E}$ and $\mathcal{C}$, we employ l2-regularization. It is noteworthy that, if we let $d_e = d_c$, and let $\mathcal{S} = \mathbf{I}$, then the alignment model degenerates to a simplified case based on semi-non-negative matrix factorization (SNMF):

$$J_{\text{A}}^{\text{SNMF}} = \left\|\mathcal{X} - \mathcal{E}\mathcal{C}^{\top}\right\|_F \text{ s.t. } \mathcal{E} \geq 0 \text{ and } \mathcal{C} \geq 0 \tag{4.13}$$

The SNMF-based alignment model does not allow genes and cells to be embedded into spaces with different dimensionalities, which is less tractable to represent scRNA-seq data where the numbers of cells and monitoring gene sequences are highly different.

4.6 Cell Clustering

We evaluate `KG-Transfer` based on the task of cell clustering.

4.6.1 Dataset

The dataset we use for experiment based on the Zeisel dataset (Zeisel et al., 2015). This dataset is the Mouse cortex and hippocampus data contained 19,972 genes and 3,005 cells. We use random normal sampling to drop out entries in the matrix, which creates five settings with the drop-out rates of 10%, 30%, 50%, 70%, and 90%. In additional, extract the gene knowledge graph for `KG-Transfer` from the String database (Szklarczyk et al., 2016). In this knowledge graph, we only keep the protein-protein interation records that are marked as experiment verified for the proteins in Zeisel dataset, which provides 992,164 typed protein-protein interaction triples for

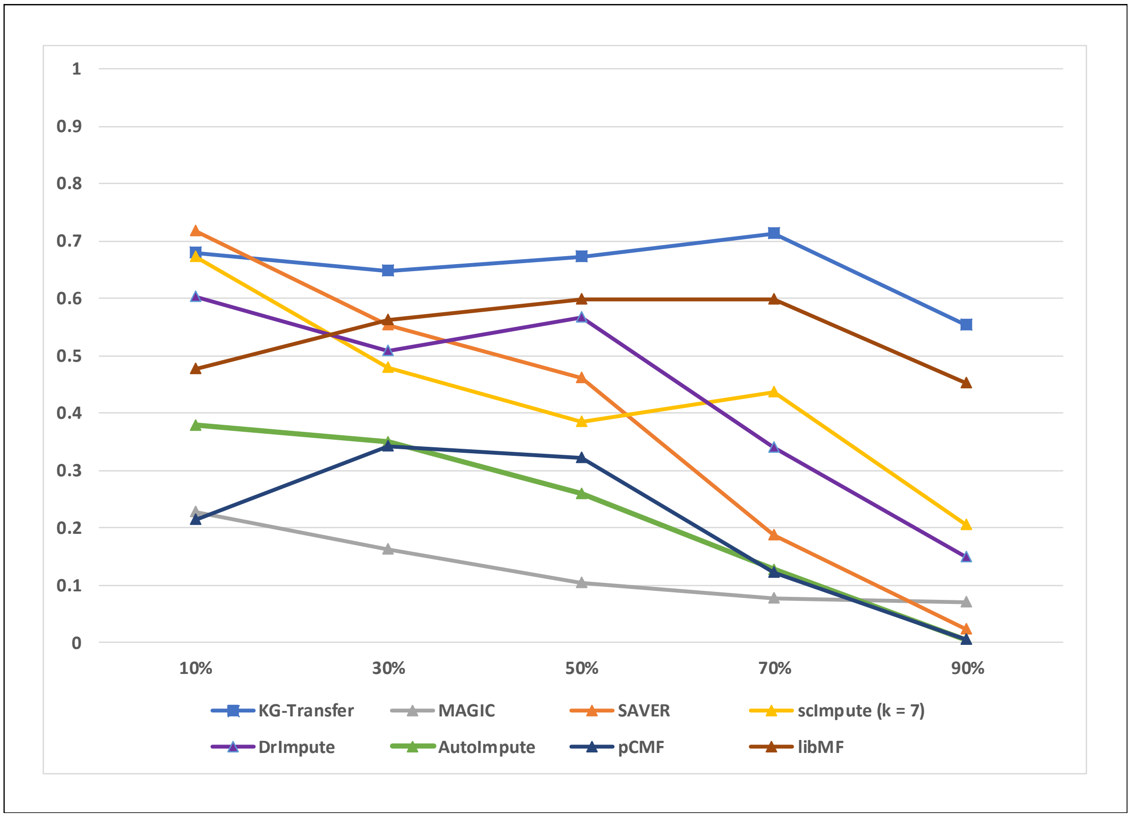


Figure 4.8: ARI of cell clustering under different drop-out rates of the Zeisel dataset.

16,084 proteins (genes).

4.6.2 Evaluation

`KG-Transfer` is compared with several recent approaches on the same task, which include SAVER (Huang et al., 2018), scImpute (Li and Li, 2018), DrImpute (Gong et al., 2018), AutoImpute (Talwar et al., 2018), MAGIC (van Dijk et al., 2017), pCMF (Durif et al., 2017) and the regular matrix factorization (libMF (Chin et al., 2016)).

After extensive hyperparameter tuning, we set the configuration of `KG-Transfer` as follows. The coefficients of the learning objective is set as $\omega_1 = 2.5$ and $\omega_2 = 0.01$. Margin $\gamma = 5.0$, and dimensionalities $d_e = 32$, $d_c = 16$. We use AMSGrad for optimization, for which the batch size is set to 256, the learning rate is set to 0.01, and exponential decay rates are set by default. Training lasts 400 epochs.

Evaluation protocols. We use Jaccard-Louvain (a.k.a. Phenograph (Levine et al., 2015)) algo-

Table 4.8: ARI of cell clustering by KG-Transfer verses the best of all baselines under different drop-out rate.

Drop-out rate	10%	30%	50%	70%	90%
KG-Transfer	0.679	**0.647**	**0.673**	**0.713**	**0.554**
Best of all baselines	**0.716**	0.561	0.599	0.597	0.452

rithm to perform clustering based on the obtained cell embeddings. This is due to that Phenograph has been identified to be the best performing clustering algorithm by previous works (Shekhar et al., 2016; Talwar et al., 2018). Following the convention, we report Adjusted Random Index (ARI) as the evaluation metric. The higher ARI indicates better clustering results.

Results The results by KG-Transfer and all the baselines are shown in Figure 4.8. Specifically, Table 4.8 compares the ARI by KG-Transfer against the best-performing baseline under each drop-out rate setting. Except for that under 10% drop-out, SAVER performs better than KG-Transfer, under all the other settings, KG-Transfer significantly outperforms all the baselines. The results also show that by transfering knowledge from the gene interaction view, KG-Transfer is very invariant to the sparsity of transcript data caused by high drop-out rate. However, for almost all baselines, except for NMF (libMF), the performance drops drastically with extreme drop-out. Since the drop-out events are typically estimated to count for 80% to 90% of the lab transcripts for scRNA-seq data, this indicates that leveraging the gene level knowledge by KG-Transfer is important for addressing this problem.

4.7 Conclusion

In this chapter, we propose two novel models to support knowledge transfer across different views of multi-relational data based on complex alignment information. JOIE aiming to jointly embed real-world entities and ontological concepts based on one-to-many alignment. We characterize a two-view knowledge base. In the embedding space, our approach jointly captures both structured knowledge of each view, and type links that bridges the two views. Extensive experiments on the tasks of knowledge graph completion and entity typing show that our model JOIE can successfully capture latent features from both views in knowledge bases, and outperforms various state-of-the-art baselines. KG-Transfer captures the fuzzy alignment between genes and cells, and transfer the gene-level knowledge to help cell clustering. Under frequent drop-out rates, `KG-Transfer` shows significant advancement over a handful of recent approaches on the same task.

CHAPTER 5

Learning to Capture Relational Properties

In this chapter, we discuss learning to preserve relational properties in the embedding space. Such relational properties are frequently existing in ontology graphs.

5.1 Introduction

Populating large ontologies has been a critical challenge to the Semantic Web. In the past decade, several well-known ontology graphs have been created and widely utilized, including Yago (Mahdisoltani et al., 2015), ConceptNet (Speer et al., 2017), and DBpedia OWL (Lehmann et al., 2015). Although some of these graphs contain millions of relation facts, they still face the coverage and completeness issues that have been the subject of much research (Mousavi et al., 2014b; Quan et al., 2004). This is because enriching such large structures of expertise knowledge requires levels of intelligence and labor that is hardly affordable to humans. Hence, some works have proposed to mine ontologies from text using parsing-based (Culotta and Sorensen, 2004; Fundel et al., 2007; Mousavi et al., 2014a) or fuzzy-logic-based (Lau et al., 2009; Quan et al., 2004; Widyantoro and Yen, 2001) techniques. However, in practice, these techniques are often limited by the lack of high-quality reference corpora that are required for the harvest of the dedicated domain knowledge. Also, the precise recognition of relation facts for the ontology is another unsolved problem, since these relation facts are very high-level and are often not explicitly expressed in the corpora (Lau et al., 2009). Hence, these methods merely help populate some small ontology graphs in narrow domains such as gene ontologies and scholarly ontologies (Cheng et al., 2004; Quan et al., 2004), but they have not been successfully used to improve the completeness of these large cross-domain ontology graphs such as Yago and ConceptNet.

While knowledge graph embedding methods help enrich instance-level knowledge graphs, previous related works only focus on capturing the simple relations in instance-level knowledge graphs, paying less attention to the complex relations in ontology graphs. In fact, relation facts in ontology graphs are often defined with relational properties, such as transitivity and symmetry, as well as form hierarchies. A typical example is provided by *Is-A*, which is both transitive and hierarchical, and is the most frequently appearing semantic relation in ontologies. We find that, in well-known ontology graphs, complex relations usually comprise the majority: 85% of the triples in Yago, 96% of the triples in ConceptNet, and 47% of the triples in DBpedia OWL enforce relational properties, while 60%, 38%, and 48% of these triples are defined with hierarchical relations. However, existing methods fail to represent these complex relations for several reasons: (i) These methods at most use the same relation-specific projection in the energy function, but fail to differentiate the components of triples. Therefore, they are ill-posed to characterize triples with relational properties. In fact, the encoding of a concept that serves as different components in such triples, i.e. either s or t, must be differentiated so as to correctly preserve relational properties in the embedding spaces (as shown in Section 5.2.2). (ii) These methods also lack a learning phase that is dedicated to hierarchical relations. This also impairs the preciseness of embeddings. We observe in our experiments that, above limitations largely hinder the effectiveness of existing methods for ontology graphs.

We propose On2Vec, a translation-based graph embedding model that specializes in characterizing the complex semantic relations. On2Vec adopts two component models: the *Component-specific Model* which preserves the relational properties by applying component-specific projections on source and target concepts respectively, and the *Hierarchy Model* which performs an attentive learning process on hierarchical relations. We evaluate our model with the tasks of relation prediction and relation verification, which respond respectively to the following two questions: (i) What relation should be added between two concepts? (ii) Is the predicted relation correct? Experimental results on data sets extracted from Yago, ConceptNet, and DBpedia OWL show promising results and significant improvement on related methods.

5.2 Embedding Ontology Graphs With Complex Relational Properties

In this section, we introduce the proposed method for learning ontology graph embeddings. We begin with the formalization of ontology graphs.

5.2.1 Preliminary

An ontology is a graph $G(C, R)$ where C is the set of concepts, and R is the set of semantic relations. $T = (s, r, t) \in G$ denotes a triple that represents a relation fact, for which $s, t \in C$ and $r \in R$. Boldfaced $\mathbf{s}$, $\mathbf{r}$, $\mathbf{t}$ respectively represent the embedding vectors of source s, relation r, and target t. Relations are further classified by $R = R_{tr} \cup R_s \cup R_h \cup R_o$, which respectively denote the sets of transitive, symmetric, hierarchical, and other simple relations. We do not specify reflexive relations here because such relations can be easily model as a zero vector by any translation-based model. R_{tr} and R_h thereof, are not required to be disjoint, while R_o is disjoint with all the rest three. For transitive relations, that is to say, given $r \in R_{tr}$, and three different concepts $c_1, c_2, c_3 \in C$, if $(c_1, r, c_2), (c_2, r, c_3) \in G$, then $(c_1, r, c_3) \in G$. As for symmetric relations, that is to say, given $r \in R_s$, and two different concepts $c_1, c_2 \in C$, if $(c_1, r, c_2) \in G$, then $(c_2, r, c_1) \in G$. As for hierarchical relations, we further divide them into $R_h = R_r \cup R_c$ where R_r denotes refinement relations that partition coarser concepts into finer ones, and R_c denotes coercion relations that group finer concepts to coarser ones (Camossi et al., 2006; Chen et al., 2016a,b).

5.2.2 Modeling

On2Vec adopts two component models that learn on the two facets of the ontology graph: the *Component-specific Model* (CSM) which encodes concepts and relations into low-dimensional embedding spaces without the loss of the relational properties, and the *Hierarchy Model* (HM) which strengthens the learning process on hierarchical relations with an auxiliary energy.

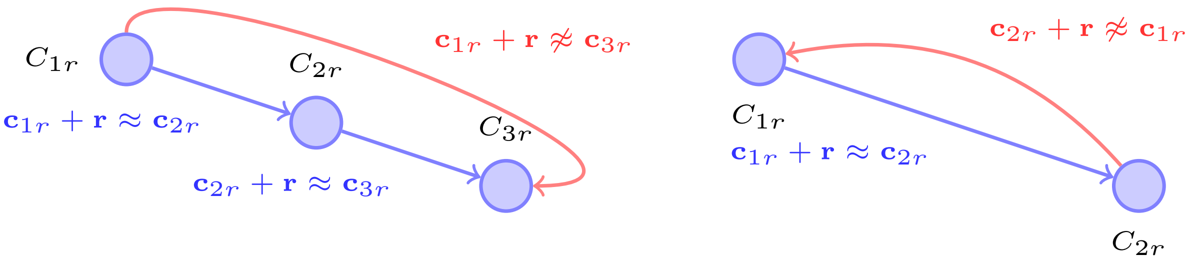

C_{1r}
C_{2r}
C_{3r}
$\mathbf{c}_{1r} + \mathbf{r} \approx \mathbf{c}_{2r}$
$\mathbf{c}_{2r} + \mathbf{r} \approx \mathbf{c}_{3r}$
$\mathbf{c}_{1r} + \mathbf{r} \not\approx \mathbf{c}_{3r}$
C_{1r}
C_{2r}
$\mathbf{c}_{1r} + \mathbf{r} \approx \mathbf{c}_{2r}$
$\mathbf{c}_{2r} + \mathbf{r} \not\approx \mathbf{c}_{1r}$

Figure 5.1: Depiction of the conflicts of the relation-specific projection for learning transitive relations (Case 1, left), and symmetric relations (Case 2, right).

5.2.2.1 Component-specific Model.

The reason that previous translation-based models fail to preserve relational properties is because the relation-specific projection f_r place concepts involved in transitive or symmetric relations at conflict positions. Fig. 5.1 depicts such conflicts, and a brief proof is given below:

- **Case 1.** Consider $r \in R_{tr}$ and $c_1, c_2, c_3 \in C$ such that (c_1, r, c_2), (c_2, r, c_3), $(c_1, r, c_3) \in G$, where c_1, c_2, and c_3 are projected to $\mathbf{c}_{1r}$, $\mathbf{c}_{2r}$, and $\mathbf{c}_{3r}$ respectively by f_r. Then if $\mathbf{c}_{1r} + \mathbf{r} \approx \mathbf{c}_{2r}$ and $\mathbf{c}_{2r} + \mathbf{r} \approx \mathbf{c}_{3r}$ hold for the first and second triples, it is impossible for $\mathbf{c}_{1r} + \mathbf{r} \approx \mathbf{c}_{3r}$ to hold for the third triple, since $\mathbf{r} \neq 0$ (otherwise $\mathbf{r}$ does not provide a valid vector translation).
- **Case 2.** Consider $r \in R_s$ and $c_1, c_2 \in C$ such that $(c_1, r, c_2), (c_2, r, c_1) \in G$, where c_1 and c_2 are projected to $\mathbf{c}_{1r}$ and $\mathbf{c}_{2r}$ respectively by f_r. Then it is not possible for both $\mathbf{c}_{1r} + \mathbf{r} \approx \mathbf{c}_{2r}$ and $\mathbf{c}_{2r} + \mathbf{r} \approx \mathbf{c}_{1r}$ to hold, since $\mathbf{r} \neq 0$.

Hence, to solve the conflicts in the above two cases, CSM provides two component-specific (and also relation-specific) projections to differentiate the encoding of the same concept that serves as different components in triples. The general form of the energy function is given as below,

$$S_d(T) = \|f_{1,r}(\mathbf{s}) + \mathbf{r} - f_{2,r}(\mathbf{t})\|$$

where $f_{1,r}$ and $f_{2,r}$ are respectively the component-specific projections for the source and the target concepts. It is easy to show that the component-specific projections are able to solve the conflicts in learning the relational properties, as c_2 in Case 1 is projected differently when it serves as the source of (c_1, r, c_2) or the target of (c_2, r, c_3), while both c_1 and c_2 in Case 2 can be learnt to be embedded in opposite positions respectively for (c_1, r, c_2) and (c_2, r, c_1) by the two projections. Corresponding conclusion can be easily extended to cases with more than three relation facts via mathematical induction.

Besides measuring the plausibility (or the opposite: dissimilarity) of a given triple, S_d is also the basis for predicting missing relation facts for an ontology. Given two concepts s and t, we find the r which leads to the lowest S_d. The forms of $f_{1,r}$ and $f_{2,r}$ are decided particularly by the techniques to differentiate the concept encoding under different contexts of relations. In this chapter, we adopt the relation-specific linear transformations (Lin et al., 2015). Hence, we have $f_{1,r}(\mathbf{s}) = \mathbf{M}_{1,r}\mathbf{s}$ and $f_{2,r}(\mathbf{t}) = \mathbf{M}_{2,r}\mathbf{t}$, such that $\mathbf{M}_{1,r}, \mathbf{M}_{2,r} \in \mathbb{R}^{k \times k}$. Other techniques like hyperplane projections, dynamic matrices, and bilinear transformations may also be considered, which we leave as future work.

The objective of CSM is to minimize the total S_d energy of all triples. To achieve more efficient learning, we import negative sampling to the learning process, which is widely applied in previous works (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015; Wang et al., 2014b). Unlike these works that select negative samples on entities (or concepts), we perform negative sampling on semantic relations to better suit our tasks. Then the complete energy function of CSM is defined as the following hinge loss,

$$\begin{aligned} S_{\text{CSM}}(G) = \sum_{(s,r,t) \in G} [&\|f_{1,r}(\mathbf{s}) + \mathbf{r} - f_{2,r}(\mathbf{t})\| \\ &- \|f_{1,r}(\mathbf{s}) + \mathbf{r}' - f_{2,r}(\mathbf{t})\| + \gamma_1]_+ \end{aligned}$$

for which r' is a randomly sampled relation that does not hold between s and t, γ_1 is a positive margin, and $[x]_+$ denotes the positive part of x (i.e., $\max(x, 0)$).

5.2.2.2 Hierarchy Model.

For a hierarchical relation, we often have multiple finer concepts that apply this relation to a coarser one. In this case, we appreciate a good representation where all the embeddings of the finer concepts converge closely in a tight neighborhood, which corresponds to low dissimilarity of the embedded relation. However, it is very likely for the learning process to spread out the embeddings of the finer concepts. Because each of the finer concepts can participate in multiple relation facts, encoding of a concept in one relation fact can be easily interfered by that of many other relation facts. This no doubt indicates low plausibility measures of the triples, and imprecise vector translation for the corresponding relations. Therefore, HM is dedicated to converge closely the projected embeddings of every finer concepts for a hierarchical relation.

To facilitate the definition of the energy function, we first define a *refine* operator denoted as σ:

- Given $r \in R_r$, $c \in C$, then $\sigma(c, r) = \{c' | (c, r, c') \in G\}$ fetches all the finer concepts c' that directly apply the refinement relation r to the coarser c.
- Given $r \in R_c$, $c \in C$, then $\sigma(c, r) = \{c' | (c', r, c) \in G\}$ fetches all the finer concepts c' that directly apply the coercion relation r to the coarser c.

The energy function of HM is defined below,

$$\begin{aligned} S_{hm}(G) = & \sum_{r \in R_r} \sum_{s \in C} \sum_{t \in \sigma(s,r)} \omega \left(f_{1,r}(\mathbf{s}) + \mathbf{r}, f_{2,r}(\mathbf{t}) \right) \\ & + \sum_{r \in R_c} \sum_{t \in C} \sum_{s \in \sigma(t,r)} \omega \left(f_{2,r}(\mathbf{t}) - \mathbf{r}, f_{1,r}(\mathbf{s}) \right) \end{aligned}$$

where ω is a function that monotonically increases w.r.t. the angle or the distance of the two argument vectors. In practice, ω can be easily implemented as cosine distance.

Negative sampling is imported to rewrite S_{hm} as below,

$$\begin{aligned} S_{\mathrm{HM}}(G) = & \sum_{r \in R_r} \sum_{s \in C} \sum_{t \in \sigma(s,r) \wedge t' \notin \sigma(s,r)} S_{hr} \\ & + \sum_{r \in R_c} \sum_{t \in C} \sum_{s \in \sigma(t,r) \wedge s' \notin \sigma(t,r)} S_{hc} \end{aligned}$$

Algorithm 2: Learning procedure of On2Vec.

```
Input: Training set $G = \{(s, r, t\}$, hyperparameters $\alpha_1$ and $\alpha_2$, learning rate $\lambda$, batch size $b$
Output: parameters $\theta$ for embedding vectors and projections
Randomly initialize $\theta$;
while training is not terminated do
    $G_{\text{CSM}} \leftarrow \mathsf{Sample}(G, b)$ ;                              /* Sample size $b$. */
    $G_{\text{HM}} \leftarrow B_{\text{CSM}} \leftarrow B_{\text{HM}} \leftarrow \emptyset$;
    while $|G_{\text{HM}}| < b$ do
        $c \leftarrow \mathsf{Sample}(c) \in C$;
        $r \leftarrow \mathsf{Sample}(r) \in R_h$;
        $G_{\text{HM}} \leftarrow G_{\text{HM}} \cup \sigma(c, r)$ ;                 /* Truncate if $|G_{\text{HM}}| \geq b$. */
    for $T(s, r, t) \in G_{\text{CSM}}$ do
        $T'(s, r', t) \leftarrow \mathsf{NegativeSample}(T)$;
        $B_{\text{CSM}} \leftarrow B_{\text{CSM}} \cup \{(T, T')\}$ ;                /* Batch for CSM. */
    for $T(s, r, t) \in G_{\text{HM}}$ do
        if $r \in R_r$ then
            /* Negative sampling for a refinement relation. */
            $T'(s, r, t') \leftarrow \mathsf{NegativeSample}(T)$ ;
        else
            /* Negative sampling for a coercion relation. */
            $T'(s', r, t) \leftarrow \mathsf{NegativeSample}(T)$ ;
        $B_{\text{HM}} \leftarrow B_{\text{HM}} \cup \{(T, T')\}$ ;                  /* Batch for HM. */
    $\theta \leftarrow \theta - \lambda \nabla S_{\text{CSM}}(B_{\text{CSM}})$;
    $\theta \leftarrow \theta - \lambda \nabla \alpha_1 S_{\text{HM}}(B_{\text{HM}})$;
    $B_c \leftarrow B_r \leftarrow \emptyset$ ;                                      /* Batch for soft-constraint. */
    for $(T, T') \in B_{\text{CSM}} \cup B_{\text{HM}}$ do
        $B_c \leftarrow B_c \cup \{s, s', t, t'\}$ ;                                  /* Concepts in triple batches. */
        $B_r \leftarrow B_r \cup \{r, r'\}$ ;                                         /* Relations in triple batches. */
    $\theta \leftarrow \theta - \lambda \nabla \alpha_2 S_N(B_c, B_r)$;
```

Table 5.1: Model complexity: number of parameters for optimization, and the computational complexity for predicting a relation. n_c and n_r are numbers of concepts and relations, and k is the dimensionality of embeddings.

Model	#Parameters	Complex. rel. predict.
TransE	$\mathrm{O}(n_c k + n_r k)$	$\mathrm{O}(k + n_r k^2)$
TransH	$\mathrm{O}(n_c k + 2n_r k)$	$\mathrm{O}((3n_c + 1)k + n_r k^2)$
TransR	$\mathrm{O}(n_c k + n_r k^2)$	$\mathrm{O}(n_c k^2 + k + n_r k^2)$
TransD	$\mathrm{O}(n_c k + 2n_r k)$	$\mathrm{O}(3n_c k^2 + k + n_r k^2)$
On2Vec	$\mathrm{O}(n_c k + 2n_r k^2)$	$\mathrm{O}((n_c + 1)k^2 + k + n_r k^2)$

Table 5.2: Statistics of the data sets. pct. prop. and pct. hier. are the percentages of triples defined with relational properties and hierachies.

Data Set	DB3.6k	CN30k	YG15k	YG60k
#trip.	6,485	286,763	219,472	522,282
pct. prop.	47.39%	96.89%	45.69%	85.58%
pct. hier.	47.11%	59.96%	76.80%	59.96%
#rel.	8	41	17	17
#con.	3,625	29,564	14,887	56,910
#train.	5,485	256,762	204,064	472,280
#valid.	500	10,001	5,000	10,000
#test.	500	20,000	10,400	40,000

such that s' and t' are negative samples of concepts, S_{hr} and S_{hc} are respectively the hinge loss for refinement and coercion relations defined as below, where γ_2 is a positive margin.

$$S_{hr} = \left[\omega\left(f_{1,r}(\mathbf{s}) + \mathbf{r}, f_{2,r}(\mathbf{t})\right) - \omega\left(f_{1,r}(\mathbf{s}) + \mathbf{r}, f_{2,r}(\mathbf{t'})\right) + \gamma_2\right]_+$$

$$S_{hc} = \left[\omega\left(f_{2,r}(\mathbf{t}) - \mathbf{r}, f_{1,r}(\mathbf{s})\right) - \omega\left(f_{2,r}(\mathbf{t}) - \mathbf{r}, f_{1,r}(\mathbf{s'})\right) + \gamma_2\right]_+$$

Table 5.1 gives the model complexity of `On2Vec` and some related models in terms of parameter sizes. We also give out the computational complexity of the relation prediction for a pair of concepts, which is the most frequent operation in our tasks. Although `On2Vec` unavoidably increases the parameter sizes due to additional projections, it keeps the computational complexity of relation prediction at the same magnitude as TransR, which is lower than TransD.

5.2.3 Learning Process

The objective of learning `On2Vec` is to minimize the combined energy of S_{CSM} and S_{HM}. Meanwhile, norm constraints are enforced on embeddings and projections to prevent training from a trivial solution where vectors collapse to infinitely large (Bordes et al., 2014a; Chen et al., 2017b; Wang et al., 2014b). Such constraints are conjuncted below.

$$\forall c \in C, \forall r \in R : \|\mathbf{c}\| \leq 1 \wedge \|f_{1,r}(\mathbf{c})\| \leq 1 \wedge \|f_{2,r}(\mathbf{c})\| \leq 1 \wedge \|\mathbf{r}\| \leq 2$$

In the learning process, these constraints are quantified as soft constraints:

$$\begin{aligned} S_{\text{N}}(C,R) = \sum_{c \in C} ([\|\mathbf{c}\| - 1]_+ + [\|f_{1,r}(\mathbf{c})\| - 1]_+ \\ +[\|f_{2,r}(\mathbf{c})\| - 1]_+) + \sum_{r \in R} [\|\mathbf{r}\| - 2]_+ \end{aligned}$$

Finally, learning `On2Vec` is realized by using batch stochastic gradient descent (SGD) (Needell et al., 2014) to minimize the joint energy function given as below,

$$J(\theta) = S_{\text{CSM}} + \alpha_1 S_{\text{HM}} + \alpha_2 S_{\text{N}}$$

where α_1 and α_2 are two non-negative hyperparameters, and θ is the set of model parameters that include embedding vectors and projection matrices. Empirically (as shown in (Lin et al., 2015; Wang et al., 2014b)), α_2 is assigned with a small value within (0, 1]. α_1 is adjusted in experiments to weigh between the two component models. Instead of directly updating J, the learning process optimizes S_{CSM} and $\alpha_1 S_{\text{HM}}$ in separated groups of batches, and the batches from both groups are used to optimize $\alpha_2 S_{\text{N}}$. We initialize vectors by drawing from a uniform distribution on the unit spherical surface, and initialize matrices using random orthogonal initialization (Saxe et al., 2014). The detailed optimization procedure is given in Algorithm 2.

5.3 Experiments

In this section, we evaluate `On2Vec` on two tasks that answer two important questions for ontology population: (i) Relation prediction: what is the relation to be added between a given pair of concepts? (ii) Relation verification: is a candidate relation fact correct or not?

The baselines that we compare against include the representative translation-based embedding methods TransE, TransH, TransR, and TransD (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015; Wang et al., 2014b), and neural methos RESCAL and HolE (Nickel et al., 2016, 2011). Experimental results are reported on four data sets extracted from DBpedia, ConceptNet, and Yago, for which complex relation types have been predefined. Statistics of the data sets are shown in Table 5.2. All the meta relations that assign URIs and system timestamps are removed during the preparation of the data sets. To simplify the experiments, transitive relations are limited to four-hops. Relation facts for extra hops are hence discarded. Since DBpedia provides both ontology and instance-level graphs, we keep only the ontology view to obtain DB3.6k. CN30k and YG15k are extracted from English versions of ConceptNet and Yago respectively. These two graphs match the number of nodes with WN18 and FB15k respectively, which are two commonly-used instance-level graphs in related works (Bordes et al., 2012, 2013, 2011; Ji et al., 2015; Lin et al., 2015; Wang et al., 2014b; Yang et al., 2015c). YG60k is a much larger data set that is about half of the entire English-version Yago after data cleaning. Each data set is randomly partitioned into training, validation, and test sets.

5.3.1 Relation Prediction

This task aims at extending an ontology graph by predicting the missing relations for given concept pairs.

Table 5.3: Accuracy of Relation Prediction (%). prop. means with properties, hier. means hierarchical relations.

Data Sets	DB3.6k			CN30k			YG15k			YG60k		
Rel Type	prop.	hier.	overall	prop.	hier.	overall	prop.	hier.	overall	prop.	hier.	overall
TransE	8.40	8.71	13.31	5.09	3.21	8.01	2.03	0.56	0.20	0.02	0.00	0.16
TransH	47.55	47.83	50.80	13.8	7.29	13.66	65.53	61.57	66.27	62.92	43.79	59.78
TransD	50.40	57.98	80.74	72.34	76.18	77.67	74.42	75.60	77.77	72.39	66.18	73.23
TransR	68.14	71.72	78.32	79.32	84.37	80.56	79.74	79.56	79.81	77.40	71.19	78.22
RESCAL	29.70	35.65	36.19	55.39	56.06	54.46	58.88	54.50	59.07	52.36	53.16	58.51
HolE	82.76	81.68	89.63	79.21	80.99	77.71	76.78	75.20	79.13	73.69	74.47	78.10
O2V w/ HM	86.46	**89.65**	**93.35**	**88.99**	**96.05**	**89.21**	**88.88**	**89.36**	**88.75**	**89.09**	**88.71**	**88.74**
O2V w/o HM	**86.85**	86.06	90.69	85.58	95.07	86.01	85.87	83.98	84.29	80.57	75.96	81.47

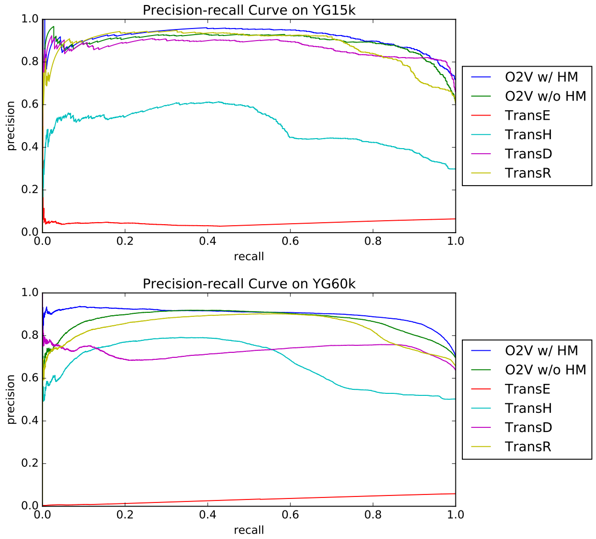


Figure 5.2: Precision-recall curves for relation prediction on YG15k and YG60k.

Evaluation Protocol. We evaluate our approach by way of held-out evaluation (Lin et al., 2016; Wang et al., 2014a). Each model is trained on the training set that represents the known ontology. Then, for each case in the test set, given the source and target concepts, the model predicts the relation that leads to the lowest dissimilarity score S_d defined in Section 5.2.2.1. To evaluate with controlled variables, on each data set, we employ the same configuration for every models. On DB3.6k, we fix dimensionality $k = 25$, margin $\gamma_1 = 2.0$, learning rate $\lambda = 0.005$, $\alpha_2 = 0.5$, and l_1 norm. CN30k and YG15k shares the configuration as $k = 50$, $\gamma_1 = 0.5$, $\lambda = 0.001$, $\alpha_2 = 0.5$, and l_2 norm. Lastly, we use $k = 100$, $\gamma_1 = 0.5$, $\lambda_1 = 0.001$, $\alpha_2 = 0.5$, and l_2 norm. $\gamma_2 = 0.5$ is configured for `On2Vec`. To test the effect of HM, we also provide two versions of `On2Vec`. One version (`On2Vec` w/ HM) is set with $\alpha_1 = 0.75$, which is empirically decided via the hyperparameter study in Section 5.3.3. The other version (On2Vec w/o HM) nullifies HM by setting $\alpha_1 = 0$. To enable batch sampling for HM, we implement the σ function for hierarchical relation facts using hash trees. The learning process is stopped once the accuracy on the validation set stops improving.

Results. The overall accuracy is reported per data set in Table 5.3. On each data set, we also aggregate respectively the accuracy on the test cases with relational properties, as well as the accuracy on those with hierarchical relations. We discover that, TransE, though has performed well on encoding instance-level knowledge graphs (Bordes et al., 2013), receives unsatisfactory results on predicting the complex ontology relations. By learning each relation type on a different hyperplane, TransH notably solves the problem of TransE, but appears to fail on CN30k where the candidate space is larger than other graphs. TransR and TransD provide more robust characterization of relations than TransH, especially in TransR where relation-specific projections are implemented as linear transformations. However, the overall performance of both TransR and TransD is impaired by the two types of complex relations. For neural models, HolE adapts better on the smaller DB3.6k data set, while it is at most comparable to TransR and TransD on larger ones, and RESCAL is less successful on all settings. As expected, On2Vec greatly outperforms the above baseline methods, regardless of whether HM is enabled or not. The On2Vec with HM thereof, outperforms the best runner-up baselines respectively in all settings by 3.72%$\sim$10.52% of overall accuracy, 4.09%$\sim$11.69% of accuracy on cases with relational properties, and 7.97%$\sim$14.24% of accuracy on cases with hierarchical relations. We also discover that, when HM is enabled, it leverages the accuracy on hierarchical relations by up to 12.75%, and overall accuracy by up to 7.27%, and does not noticeably cause interference to the prediction for cases with relational properties. Though, the advantage of CSM alone (i.e. On2Vec w/o HM) is still significant over the baselines. Since the relation prediction accuracy of On2Vec is close to 90% on all four data sets, this indicates that On2Vec achieves a promising level of performance in populating ontology graphs, and it is effective on both small and large graphs.

We also perform precision-recall analysis on the two Yago data sets on translation-based models. To do so, we calculate the dissimilarity scores S_d (Equation 5.2.2.1) for the possible predictions

Table 5.4: Accuracy of relation verification (%). prop. means with properties, hier. means hierarchical relations.

Data Sets	DB3.6k			CN30k			YG15k			YG60k		
Rel Type	prop.	hier.	overall	prop.	hier.	overall	prop.	hier.	overall	prop.	hier.	overall
TransE	67.49	71.44	67.57	69.14	18.23	51.85	58.73	62.69	69.09	60.92	61.30	66.89
TransH	72.88	82.06	69.71	93.40	86.16	94.17	69.24	72.96	89.20	66.47	71.62	88.81
TransD	76.79	81.11	74.44	91.63	84.20	93.36	65.63	70.58	88.01	61.76	71.08	86.34
TransR	77.11	86.82	73.76	85.83	52.01	74.73	71.80	72.73	88.63	71.92	71.09	87.77
RESCAL	75.30	74.61	76.20	70.41	75.64	72.28	68.76	67.30	72.29	69.36	69.16	76.21
HolE	82.89	79.23	85.90	90.31	91.43	91.18	78.31	77.10	86.88	71.22	70.80	87.67
O2V w/ HM	**95.46**	**94.97**	**95.57**	97.19	**95.54**	**98.04**	80.33	**78.39**	**93.29**	**74.92**	**73.36**	**91.79**
O2V w/o HM	91.94	91.15	91.74	**97,99**	93.73	96.51	**81.01**	74.30	91.12	73.72	72.93	90.97

of each test case, and select those that are not ranked behind the correct prediction. Then a threshold is initiated as the minimum dissimilarity score. The answer set is inserted with predictions for which the dissimilarity scores fall below the threshold, and the answer set grows along with the increasing of the threshold, until all correct predictions are inserted. Therefore, we obtain the precision-recall curves in Fig. 5.2, for which the area under curve is reported as: (i) For YG15k, On2Vec w/ HM: **0.9138**; On2Vec w/o HM: 0.8938; TransE: 0.0457; TransH: 0.4973; TransD: 0.8386; TransR: 0.8587. (ii) For YG60k, On2Vec w/ HM: **0.9005**; On2Vec w/o HM: 0.8703; TransE: 0.0313; TransH: 0.6688; TransD: 0.7275; TransR: 0.8372. This further indicates that On2Vec achieves better performance than other baselines, and HM improves the performance of On2Vec with CSM alone.

5.3.2 Relation Verification

Relation verification aims at judging whether a relation marked between two concepts is correct or not. It produces a classifier that helps to verify the candidate relation facts.

Evaluation Protocol. Because this is a binary classification problem that needs positive and negative cases, we use a complete data set as the positive cases. Then, following the approach of (Socher et al., 2013), we corrupt the data set to create negative cases. In detail, a negative case is created by (i) randomly replacing the relation of a positive case with another relation, or (ii) randomly assign a relation to a pair of unrelated concepts. Options (i) and (ii) respectively contribute negative cases that are as many as 100% and 50% of positive cases. We perform a 10-fold cross-validation. Within each fold, embeddings and the classifier are trained on the training data, and the classifier is evaluated on the remaining validation data.

Table 5.5: Examples of top-ranked new relation facts. The italic ones are conceptually close. The rest are correct.

CN30k
<Offer, Entails, Degree>
<Offer, Entails, Decide>
<State, IsA, Boundary>
<National_Capital, IsA, Boundary>
<Get_in_line, HasFirstSubevent, Pay>
<Convert, SimilarTo, Transform>
<Person, ReceivesAction, Hint>
<Stock, Entails, Receive>
<Evasion, HasContext, Physic>
YG60k
<Luisa_de_Guzmán, isMarriedTo, John_IV_of_Portugal>
<Georgetown, isLocatedIn, South_Carolina>
<Gmina_pomiechówek, isLocatedIn, Gmina_Konstancin>
<Örebro_Airport, isLocatedIn, Karlskoga>
<Horgen, isLocatedIn, Bülach_District>
<Luxor_International_Airport, isConnectedTo, Daqing_Sartu_Airport>
<Akron, isLocatedIn, Ohio>
<Curtis_guild_Jr, hasGender, Male>
<Aalbach, isLocatedIn, Europe>

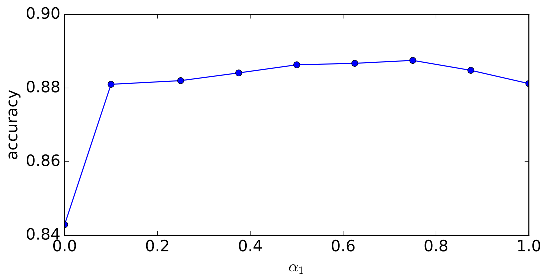


Figure 5.3: Choices of α_1 values and corresponding accuracy of relation prediction on YG15k.

We use a threshold-based classifier, which is similar to the one for triple alignment verification in (Chen et al., 2017b). This simple classifier adequately relies on how precisely each model preserves the structure of the ontology graph in the embedding space. In detail, for each case, we calculate its dissimilarity score S_d (Section 5.2.2.1). The classifier then finds a threshold τ such that $S_d < \tau$ implies positive, otherwise negative. The value of τ is determined to maximize the accuracy on the training data of each fold.

We carry forward the corresponding configurations from the last experiment, in order to show the performance of each model under controlled variables.

Results. We aggregate the mean accuracy for the two categories of complex relation facts as well as the overall accuracy for each setting. The results are shown in Table 5.4, which has a maximum standard deviation of 0.005 in cross-validation for each setting. Thus, the results are statistically sufficient to reflect the performance of classifiers. Both versions of On2Vec again outperform the other models, especially on complex relation facts. On all four data sets, On2Vec outperforms the best runner-up baselines by 2.98%$\sim$9.67% of overall accuracy, 2.02%$\sim$12.57% of accuracy for cases with relational properties, and 1.29$\sim$8.15% of accuracy on hierarchical relations. This indicates that On2Vec precisely encodes the ontology graph structures, and provides much accurate plausibility measurement to decide the correctness of unknown triples. We also discover that, On2Vec trained with HM has a drop of accuracy for up to 0.8% on cases with relational properties from CN30k and YG15k. This is likely due to that the auxiliary learning process for hierarchical relations causes minor interference to the characterization of relational properties, while HM leverages the accuracy on hierarchical relations of these two data sets by at least 1.81%, and the overall accuracy by 0.82%$\sim$3.83%. This indicates that HM is helpful in relation verification.

5.3.3 Case Study

Lastly, we provide some case studies on hyperparameter values, and some examples of relation prediction.

5.3.3.1 Hyperparameter study

We examine the hyperparameter α_1, which is the trade-off between CSM and HM. The result based on relation prediction on YG15k is shown in Fig. 5.3. As we can see, although enabling HM with even a small value of α_1 can noticeably leverage the performance of On2Vec, the influence of different values of α_1 is not very notable, and the accuracy does not always go up along with the higher α_1. In practice, α_1 may be fine-tuned for marginal improvement, while $\alpha_1 = 0.75$ can be empirically selected.

5.3.3.2 Examples of relation prediction

Relation prediction is also performed for the complete data set of CN30k and YG60k. To do so, we randomly select 20 million pairs of unlinked concepts from these two data sets, and rank all the predictions based on the dissimilarity score S_d. Then top-ranked predictions are selected. Human evaluation is used in this procedure, since there is no ground truth for the relation facts that are not pre-existing. Like previous works (Lin et al., 2016; Zeng et al., 2015), we aggregate $P@200$, i.e. the precision on the 200 predictions with highest confidence, which results in 73% and 71% respectively. Some examples of top-ranked predictions are shown in Table 5.5.

5.4 Conclusion

This chapter proposes a greatly improved translation-based graph embedding method that helps ontology population by way of relation prediction. The proposed `On2Vec` model can effectively address the learning issues on the two categories of complex semantic relations in ontology graphs, and improves previous methods using two dedicated component models. Extensive experiments on four data sets show promising capability of `On2Vec` on predicting and verifying relation facts.

The results here are very encouraging, but we also point out opportunities for further work and improvements. In particular, we should explore the effects of other possible forms of component-specific projections, such as dynamic mapping matrices and bilinear mappings. Encoding other information such as the domain and range information of concepts may also improve the precision of our tasks. More advanced applications may also be developed using `On2Vec` such as ontology-boosted question answering.

CHAPTER 6

Embedding Uncertain Knowledge Graphs

Uncertain knowledge graphs associate every relation facts with a confidence score that represents the likelihood of a relation fact. Examples of such knowledge graphs include commonsense knowledge graphs Probase (Wu et al., 2012) and NELL (Mitchell et al., 2018), and biological knowledge graphs STRING (Szklarczyk et al., 2016) and SKEMPI (Moal and Fernández-Recio, 2012). In this chapter, we propose a representation learning method for uncertain knowledge graphs (Chen et al., 2019c).

6.1 Introduction

While current methods focus on embedding deterministic knowledge, it is critical to incorporate uncertainty information into knowledge sources for several reasons. First, uncertainty is the nature of many forms of knowledge. An example of naturally uncertain knowledge is the interactions between proteins. As molecular reactions are random processes, biologists label the confidence protein-protein interactions with the evidence for the occurrence of interactions, and represent them as uncertain knowledge graphs called protein-protein interaction (PPI) graphs. Second, uncertainty enhances inference in knowledge-driven applications. For example, short text understanding often entails interpreting real-world concepts that are ambiguous or intrinsically vague. The probabilistic knowledge graph Probase (Wu et al., 2012) provides a prior probability distribution of concepts for different English terms, and such probabilistic representations have critically supported short text understanding tasks involving disambiguation. (Wang and Wang, 2016; Wang et al., 2015).

Besides, uncertain knowledge representations have benefited various other applications, such as question answering(Yih et al., 2013) and named entity recognition (Ratinov and Roth, 2009).

To capture the quantified uncertainty information with multi-relational embeddings remains an unresolved problem. This is a non-trivial task for several reasons. First, to capture uncertainty, the embeddings need to encode additional information, as the deterministic knowledge graph embedding methods only reflects if a relation exists between entities . Second, while deterministic knowledge graph embedding models target at minimizing the estimated probability of false triples via negative sampling to enhance model learning, there is no clear border between observed low-confidence relation facts and unseen relation facts. Embedding models for deterministic knowledge graphs assume all unseen relation facts are in false beliefs. Therefore, they maximize the probability of observed training cases, and minimize the probability of unseen relation facts. However, since knowledge graphs are far from complete, unseen relation facts can represent unknown positive cases. This problem is especially significant to uncertain knowledge graphs. Existing techniques hence fall short at differentiating low-confidence relation facts from unseen relation facts.

To address the above issues, we propose a new embedding model called UKGE (Uncertain Knowledge Graph Embedding), which learns embeddings of entities and relations on uncertain knowledge graphs according to confidence scores. To enhance the precision of UKGE for predicting the uncertainty of unseen relation facts, we incorporate *probabilistic soft logic* into the learning process, which seeks to propagate the confidence information of unseen relation facts. We define three variants of UKGE that differ in the mappings from the triple plausibility estimation to confidence scores. We conducted extensive experiments on three real-world uncertain knowledge graphs for three tasks: (i) *confidence prediction* seeks to predict confidence scores of unseen relation facts; (ii) *relation fact ranking* focuses on retrieving tail entities for the query $(h, r, \underline{?t})$, and ranking these retrieved tails in the right order; (iii) *relation fact classification* decides whether a given relation fact is a "strong" relation fact with high confidence.

Our models consistently outperform the baseline models.

6.2 Related Work

To the best of our knowledge, there has been no previous work on learning embeddings for uncertain knowledge graphs. We hereby discuss the next besides deterministic knowledge graph embedding methods that have been discussed in Section 2.1, we discuss the next two lines of work that are closely related to this topic.

Uncertain Knowledge Graphs An uncertain knowledge graph provides a confidence score along with every relation fact. The development of relation extraction and crowdsourcing in recent years enabled the construction of large-scale uncertain knowledge bases. ConceptNet (Speer et al., 2017) is a multilingual uncertain knowledge graph for commonsense knowledge that is collected via crowdsourcing. The confidence scores in ConceptNet mainly come from the co-occurrence frequency of the labels in crowdsourced task results. Probase (Wu et al., 2012) consists of an universal probabilistic taxonomy that is built by relation extraction. Every fact in Probase is associated with a joint probability $P_{isA}(x, y)$. NELL (Mitchell et al., 2018) collects relation facts from reading web pages, and learns their confidence scores from semi-supervised learning with Expectation-maximum (EM) algorithm. Aforementioned uncertain knowledge graphs have enabled numerous knowledge-driven applications. For example, Wang and Wang (Wang and Wang, 2016) utilize Probase to help understand short texts.

One recent work has proposed a matrix-factorization-based approach to embed uncertain networks (Hu et al., 2017). However, it cannot be generalized to embed uncertain knowledge graphs, as the model only considers the node proximity in such networks without explicit relations, and only generates the node embeddings. As far as we know, we are among the first to study the uncertain knowledge graph embedding problem.

Probabilistic Soft Logic Probabilistic soft logic (PSL) (Kimmig et al., 2012) is a framework for probabilistic reasoning. A PSL program consists of a set of first-order logic rules with conjunctive bodies and single literal heads. PSL takes the confidence from interval $[0, 1]$ as the *soft truth values* for every atom. It uses *Lukasiewics t-norm* (Lukasiewicz and Straccia, 2008) to determine to which degree a ground rule is satisfied. PSL is widely used for Most Probable Explanation

(MPE) inference and Maximum a Posteriori (MAP) inference on Hinge-Loss Markov Random Field (HL-MRF) (Bach et al., 2013). PSL, in combination with HL-MRF, are widely used in probabilistic reasoning tasks, such as social-trust prediction and preference prediction (Bach et al., 2013, 2017). In this work, we adopt PSL to support the inference for unseen relation facts.

6.3 Problem Definition

We first provide the definition of uncertain knowledge graphs.

Definition 6.1. ***Uncertain Knowledge Graphs.*** *An uncertain knowledge graph consists of a set of weighted relation facts $\mathcal{G} = \{(l, s)\}$. For each pair (l, s), $l = (h, r, t)$ is a relation fact where $h, t \in \mathcal{E}$ (the set of entities) and $r \in \mathcal{R}$ (the set of relations), and $s \in [0, 1]$ represents the confidence for this relation fact to be true.*

Some examples of weighted relation facts are listed as below.

Example 6.3.1. Weighted relation facts.
(iPod,`isA`,device):1.00
(college,`synonym`,university):0.99
(university,`synonym`,institute):0.86
(fork, `atlocation`, kitchen): 0.4

Definition 6.2. ***Uncertain Knowledge Graph Embeddings.*** *Given an uncertain knowledge graph $\mathcal{G}$, the uncertain knowledge graph embedding model aims to encode each entity and relation in a low-dimensional space. In the embedding space, the confidence of relation facts are preserved.*

Notation-wise, boldfaced $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ represent the embedding vectors of head h, relation r and tail t respectively for each relation fact. $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$ all lie in $\mathbb{R}^k$.

6.4 Modeling

In this section, we propose our model for uncertain knowledge graph embeddings. The proposed model `UKGE` encodes the knowledge graph structure according to the confidence of relation facts, such that the embeddings of relation facts with higher confidence scores receive higher plausibility values.

We first define the relation fact plausibility, then introduce how we apply probabilistic soft logic to inferring confidence scores for unseen relations in Section 6.4.2, followed by the learning process and three model variants.

6.4.1 Modeling Plausibility for Relation Facts

Definition 6.3. ***Plausibility.*** *Given a relation fact l, the plausibility $g(l) \in [0, +\infty)$ measures how likely this relation fact holds. The higher plausibility value corresponds to the higher confidence score s. We regard confidence scores as normalized plausibility values.*

Given a relation fact $l = (h, r, t)$ and their embeddings $\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}$, we infer the plausibility of (h, r, t) by the following function:

$$g(l) = \boldsymbol{r} \cdot (\boldsymbol{h} \circ \boldsymbol{t}) \tag{6.1}$$

where $\circ$ is the element-wise product, and $\cdot$ is the inner product. The intention of this function is to capture the relatedness between embeddings $\boldsymbol{h}$ and $\boldsymbol{t}$ under the condition of relation r. We employ this relation fact modeling technique for two reasons: (i) It complies with the nature of our model to quantify the confidence of an uncertain relation fact in the manner of conditional probability. (ii) It does not introduce additional parameter complexity to the model like other techniques, such as TransH (Wang et al., 2014b), TransR (Lin et al., 2015), ConvE (Dettmers et al., 2018) and ProjE (Shi and Weninger, 2017).

6.4.1.1 From plausibility to confidence scores

We predict the confidence score of a relation fact according to the inferred plausibility. We adopt a monotonically increasing function to map a plausibility value to the confidence score. We denote the ground truth confidence score of a relation fact l as $s(l)$, and the inferred confidence score as $f(l)$.

$$f(l) = \phi(g(l)), \phi : \mathbb{R} \to [0, 1] \tag{6.2}$$

In this work, we propose two choices of mapping ϕ from plausibility to confidence scores.

Logistic function. One way to map plausibility estimation to confidence scores is via a logistic function. A logistic function takes an input from $\mathbb{R}$, and its output is monotonically increasing from 0 to 1:

$$\phi(x) = \frac{1}{1 + e^{-(\mathbf{w}x + \mathbf{b})}} \tag{6.3}$$

Bounded rectifier. Another option is to employ a bounded rectifier (Chen et al., 2015):

$$\phi(x) = \min(\max(\mathbf{w}x + \mathbf{b}, 0), 1) \tag{6.4}$$

where $\mathbf{w}$ and $\mathbf{b}$ are a weight and a bias respectively.

6.4.2 Inferring Confidence Scores for Unseen Relation Facts

One major challenge of learning embeddings for uncertain knowledge graphs is to properly estimate the uncertainty of unseen relation facts. Deterministic knowledge graph embedding methods assume that all the unseen relation facts are false beliefs, and minimize their probabilities. On uncertain knowledge graphs, numerous relation facts are endowed with low confidence scores, existing techniques hence fall short at differentiating low-confidence relation facts from unseen relation facts. Simply adopting the negative sampling strategies in existing approaches is problematic to the capture of unseen relation facts.

To enhance the plausibility learning of our models and preserve the graph structure better, we introduce probabilistic soft logic (PSL) (Kimmig et al., 2012) into uncertain knowledge graph embedding. PSL is a framework for confidence reasoning. It specifies how confidence scores propagate through the relational structure by annotated rules, and determines to what extent the rules have been satisfied.

6.4.2.1 Probabilistic Soft Logic

In this subsection we introduce probabilistic soft logic (PSL), and how we apply PSL to our embedding model. A PSL program consists of a set of first order logic rules that describe logical dependencies. A logical rule is an template with placeholders for reasoning.

Example 6.4.1. Logical Rule:

($\underline{\text{A}}$,`synonym`,$\underline{\text{B}}$) $\wedge$ ($\underline{\text{B}}$,`synonym`,$\underline{\text{C}}$) $\rightarrow$ ($\underline{\text{A}}$,`synonym`,$\underline{\text{C}}$)

This logical rule describes the transitivity of the relation `synonym`, where $\underline{\text{A}}$, $\underline{\text{B}}$ and $\underline{\text{C}}$ are placeholders for entities, ($\underline{\text{A}}$,`synonym`,$\underline{\text{B}}$) $\wedge$ ($\underline{\text{B}}$,`synonym`,$\underline{\text{C}}$) is the body of the rule, and ($\underline{\text{A}}$,`synonym`,$\underline{\text{C}}$) is the head of the rule. PSL associates every atom with a *soft truth value*, from the range $[0, 1]$, which is equivalent to the confidence score in our context, instead of a deterministic Boolean value. This feature enables fuzzy reasoning.

We treat the uncertain relation facts from uncertain knowledge graphs as *ground atoms*. In our embedding learning, ground atoms in the rule body always come from the uncertain knowledge graph. We can *ground out* a logical rule and generate *ground rules* using ground atoms. Considering the Example 6.4.1 and uncertain relation facts from the Example 6.3.1, we have the following ground rule.

Example 6.4.2. Ground Rule:

(college, `synonym`, university) $\wedge$ (university, `synonym`, college) $\rightarrow$ (college, `synonym`, institute)

The assignment process of soft truth values is called an *interpretation*. We denote the soft truth values of an atom l assigned by the interpretation I as $I(l)$. We assign to $I(l)$ the ground truth confidence scores for observed relation facts, and the predicted confidence score for unseen relation facts as follows.

$$\begin{aligned} I(l) &= s_l, (l, s_l) \in \mathcal{G} \\ I(l) &= f(l), l \in \mathcal{G}^- \end{aligned} \tag{6.5}$$

PSL uses the *Lukasiewicz t-norm* to determine to which degree a rule is satisfied. Lukasiewicz t-norm provides reasoning for the logical conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$), and implication (→) as follows:

$$l_1 \wedge l_2 = \max\{0, I(l_1) + I(l_2) - 1\} \tag{6.6}$$

$$l_1 \vee l_2 = \min\{I(l_1) + I(l_2), 1\} \tag{6.7}$$

$$\neg l_1 = 1 - I(l_1) \tag{6.8}$$

$$l_1 \rightarrow l_2 = \min\{1, 1 - I(l_1) + I(l_2)\} \tag{6.9}$$

PSL considers a rule γ as *satisfied* when the truth value of its head $I(\gamma_{head})$ is the same or higher than its body $I(\gamma_{body})$. Then a rule's *distance to satisfaction* is defined as the following rectified function:

$$d_\gamma = \max\{0, 1 - p_\gamma\} \tag{6.10}$$

Consider the Example 6.4.2. We denote (college, `synonym`, university) as l_1, (university, `synonym`, college) as l_2, and (college, `synonym`, institute) as l_3. According to Equations (6.5), (6.6), and (6.9) the distance to satisfaction is calculated as below.

$$\begin{aligned} d_\gamma &= \max\{0, I(l_1 \wedge l_2) - I(l_3)\} \\ &= \max\{0, s_{l_1} + s_{l_2} - 1 - f(l_3)\} \\ &= \max\{0, 0.85 - f(l_3)\} \end{aligned}$$

where $S(l_1)$ and $S(l_2)$ are the ground truth confidence scores of corresponding relation facts in the uncertain knowledge graph.

This equation indicates that the ground rule in Example 6.4.2 is completely satisfied when $f(l_3)$, the inferred confidence score of `(college, synonym institute)`, is above 0.85. When $f(l_3)$ is below 0.85, the smaller $f(l_3)$ leads to the larger loss. We can also see that the loss is only related to the confidence of the rule head when all ground atoms in the rule body are from uncertain knowledge graphs, and their soft truth value are set as constants. In the above example, the hinge-loss over this ground rule depends on $f(l_3)$.

Specially, we add a rule to penalize the predicted confidence scores of unseen relation facts.

For an unseen relation fact $l' = (h', r', t') \in \mathcal{G}^-$, we have a ground rule γ_0:

$$\gamma_0 : \neg(h', r', t') \tag{6.11}$$

According to Equation (6.8) and (6.10), d_{γ_0} is derived as:

$$d_{\gamma_0} = \max\{1, f(l')\} \tag{6.12}$$

6.4.3 Embedding Uncertain Knowledge Graphs

In this subsection, we present the learning process of uncertain knowledge graph embeddings.

6.4.3.1 Learning on observed relation facts

Our objective is to learn representations of entities and relations that best explain $\mathcal{G}$. We first define our loss on the observed relation facts. We denote $\mathcal{S}^+$ as the set of existing relation facts in $\mathcal{G}$, $\mathcal{G}^-$ as the set relation facts that are not observed. For each uncertain relation fact $(l, s_l) \in \mathcal{G}$, we compute the mean square error (MSE) between the ground truth confidence score s_l and our prediction $f(l)$. Then loss over the observed relation facts $\mathcal{S}^+$ is calculated as below.

$$\mathcal{L}^+ = \sum_{(l,s_l)\in\mathcal{G}} |f(l) - s_l|^2 \tag{6.13}$$

6.4.3.2 Learning on unseen relation facts

In our work, we do not directly adopt PSL to infer confidence scores. Instead, we adopt the hinge-loss to evaluate to what degree our embedding model fits the annotated rules, and adjusts the embedding models accordingly.

We define the squared hinge-loss of our model over unseen relation facts.

$$\mathcal{L}^- = \sum_{l'\in\mathcal{G}^-} \sum_{\gamma\in\Gamma_{l'}} |\psi_\gamma(f(l'))|^2 \tag{6.14}$$

where $\Gamma_{l'}$ is the set of rules related to l', $\psi_\gamma(f(l'))$ represents d_γ as a function of $f(l')$. We use the squared hinge-loss to match the form of Equation (6.13). When l' is related to only $\gamma_0 : \neg(h', r', t')$, we have$\sum_{\gamma \in \Gamma_{l'}} |\psi_\gamma(f(l'))|^2 = |f(l')|^2$.

6.4.3.3 Embedding learning

Combining Equation (6.13) and (6.14), we obtain the following objective function.

$$\mathcal{L} = \sum_{(l,s)\in\mathcal{G}} \sum_{l'\in S'_l} |f(l) - s|^2 + \sum_{\gamma\in\Gamma_{l'}} |\psi_\gamma(f(l'))|^2 \tag{6.15}$$

where S'_l is the sample set obtained by corrupting positive instance $l = (h, r, t)$:

$$\begin{aligned} S'_{(h,r,t)} =& \{(h', r, t)|h' \in \mathcal{E}, (h', r, t) \in \mathcal{G}^-\} \\ & \cup \{(h, r, t')|t' \in \mathcal{E}, (h, r, t') \in \mathcal{G}^-)\} \end{aligned} \tag{6.16}$$

where $\mathcal{E}$ is the set of entities.

Particularly, when there are no rules other than γ_0 (6.11), the objective function is:

$$\mathcal{L} = \sum_{(l,s)\in\mathcal{G}} \sum_{l'\in S'_l} |f(l) - s|^2 + |f(l') - 0|^2 \tag{6.17}$$

6.4.4 Model variants

We give three variants of $\mathcal{L}$ that differ in the choices of function $f(l)$, i.e. the mapping ϕ from plausibility $g(l)$ to confidence scores. A simple choice of ϕ is a bounded rectifier, as indicated in Equation (6.4).

$$\begin{aligned} \mathcal{L}_1 = & \sum_{(l,s)\in\mathcal{G}} \sum_{l'\in S'_l} |\min(\max(\mathbf{w}(\boldsymbol{r} \cdot (\boldsymbol{h} \circ \boldsymbol{t})) + \mathbf{b}, 0), 1) - s|^2 \\ & + \sum_{\gamma\in\Gamma_{l'}} |\psi_r(\min(\max(\mathbf{w}(\boldsymbol{r'} \cdot (\boldsymbol{h'} \circ \boldsymbol{t'})) + \mathbf{b}, 0), 1))|^2 \end{aligned} \tag{6.18}$$

The above model bounds $f(l)$ to the interval $[0, 1]$ during training. Alternatively, in the second model variant, we do not bound $f(l)$ during the training phase. We still adopt Equation (6.4) to predict confidence scores, though.

$$\begin{aligned}\mathcal{L}_2 = &\sum_{(l,s)\in\mathcal{G}}\sum_{l'\in S'_{l'}} |\mathbf{w}(\boldsymbol{r}\cdot(\boldsymbol{h}\circ\boldsymbol{t}))+\mathbf{b}-s|^2 \\ &+\sum_{\gamma\in\Gamma_{l'}} |\psi_\gamma(\mathbf{w}(\boldsymbol{r}'\cdot(\boldsymbol{h}'\circ\boldsymbol{t}'))+\mathbf{b})|^2\end{aligned} \tag{6.19}$$

Lastly, in the third variant, we use logistic function to map plausibility value to confidence scores, which has the function as below.

$$\begin{aligned}\mathcal{L}_3 = &\sum_{(l,s)\in\mathcal{G}}\sum_{l'\in S'_{l}} |\sigma(\boldsymbol{r}\cdot(\boldsymbol{h}\circ\boldsymbol{t}))-s|^2 \\ &+\sum_{\gamma\in\Gamma_{l'}} |\psi_\gamma(\sigma(\boldsymbol{r}'\cdot(\boldsymbol{h}'\circ\boldsymbol{t}')))|^2\end{aligned} \tag{6.20}$$

By minimizing the objective functions, the embedding models are enabled to model the uncertain semantic relations between entities. We denote the three variants using objective functions $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ as UKGE$_1$, UKGE$_2$, UKGE$_3$ respectively.

Example 6.1 (Ground atoms)**.** *(college, synonym, university, 0.99)* *(university, synonym, institute, 0.86)*

Uncertain knowledge embedding model

As in previous work on knowledge graph embedding (Bordes et al., 2013; Yang et al., 2015b), we introduce negative sampling during model training. We assume the confidence scores of negative samples should be equal to zero. The objective function is defined as:

6.4.4.1 Rule grounding and Soft Sampling.

HL-MRF is widely applied to Most Probable Explanation (MPE) inference (Bach et al., 2013). Starting from a set of ground truth atoms and a set of first-logic rules, MPE inference seeks one interpretation over all possible atoms that satisfies the rules best.

6.4.5 Confidence score prediction

To satisfy the range constraint $c \in [0, 1$, we adopt a monotonically increasing function to map a relation fact's plausibility to its confidence score:

$$g(h, r, t) = \phi(f(h, r, t)) = \phi(\boldsymbol{r} \cdot (\boldsymbol{h} \circ \boldsymbol{t})), \phi : \mathbb{R} \rightarrow [0, 1] \tag{6.21}$$

In this chapter we propose two choices of ϕ.

- **Logistic Regression** One way to map the plausibility to confidence score is using logistic regression. We adopt sigmoid function as the mapping ϕ. Sigmoid functions have domain of all real numbers and return value monotonically increasing from 0 to 1:

  $$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6.22}$$

  Then we have the scoring function,

  $$g(h, r, t) = \sigma(f(h, r, t)) = \frac{1}{1 + exp(-\boldsymbol{r} \cdot (\boldsymbol{h} \circ \boldsymbol{t}))} \tag{6.23}$$

- **Score bound** We can also bound the plausibility measure $f(h, r, t)$ to $[0, 1]$, and take it as the confidence score:

  $$g(h, r, t) = min(max(f(h, r, t), 0), 1) \tag{6.24}$$

6.5 Experiments

In this section, we evaluate our model on three tasks, relation fact confidence prediction, relation fact ranking, and relation fact classification.

6.5.1 Datasets

Our evaluation is conducted on datasets extracted from ConceptNet, NELL, and STRING. Correspondingly, the datasets are named CN15k, NL27k and PPI5k. CN15k matches the number of

Dataset	#Ent.	#Rel.	#Rel. Facts	Avg(s)	Std(s)
CN15k	15,000	36	241,158	0.629	0.232
NL27k	27,221	404	175,412	0.797	0.242
PPI5k	5,000	6	271,666	0.415	0.213

Table 6.1: Statistics of the extracted datasets used in this chapter. Avg(s) and Std(s) thereof are the average and standard deviation of confidence scores.

nodes (about 15,000) with FB15k - the widely used benchmark dataset for deterministic knowledge graph embeddings (Bordes et al., 2013; Wang et al., 2014b; Yang et al., 2015b), while NL27k is a larger dataset. PPI5k is a denser graph with fewer entities but much more relation facts. Table 6.1 gives the statistics for the datasets. We split each dataset into three parts: 85% for training, 7% for validation, and 8% for testing.

We heuristically created and validated 6 logical rules for CN15k, 12 logical rules for NL27k, and 1 rule for PPI5k. Table 6.2 gives some examples of the logical rules.

Dataset	Logical Rules
CN15k	(A, `hasprerequisite`, B)∧(B, `hasprerequisite`, C)→(A, `hasprerequisite`, C) (A, `causes`, B)∧(B, `causes`, C)→(A, `causes`, C)
NL27k	(A, `competeswith`, B)∧(B, `competeswith`, C)→(A, `competeswith`, C) (A, `synonym`, B)∧(B, `synonym`, C)→(A, `synonym`, C)
PPI5k	(A, `binding`, B)∧(B, `binding`, C)→(A, `binding`, C)

Table 6.2: Examples of logical rules

CN15k CN15k is a subgraph induced from ConceptNet, a multilingual commonsense knowledge graph. This subgraph contains 15,000 entities and 241,158 uncertain relation facts in English. The original scores in ConceptNet vary from 0.1 to 22, where 99.6% are less than or equal to 3.0. We normalize the scores to [0,1] using a sigmoid function. For normalization, we first bound confidence scores to $[0.1, 3.0]$, and then apply biased min-max normalization into [0.1, 1].

Dataset	CN15k		NL27k		PPI5k	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE
URGE	0.063	0.201	0.037	0.148	0.007	0.053
UKGE$_1$	0.039	0.152	0.027	0.125	0.005	0.034
UKGE$_2$	**0.039**	**0.152**	**0.026**	**0.122**	**0.005**	**0.033**
UKGE$_3$	0.043	0.159	0.036	0.148	0.007	0.039

Table 6.3: Mean square error (MSE) and mean absolute error (MAE) of relation fact confidence prediction.

Dataset	head	relation	tail	confidence	predicted confidence
CN15k	choir	relatedto	sing	0.893	0.833
	risk	isA	venture	0.893	0.965
	cool	antonym	passionate	0.709	0.635
NL27k	Treasury	agentcontrols	money	0.884	0.910
	Gmail	competeswith	Hotmail	0.802	0.719
	Japan	countrylovessports	baseball	0.973	0.997

Table 6.4: Examples of confidence prediction using $\texttt{UKGE}_2$.

Dataset	CN15k	NL27k	PPI5k
TransE	0.035	0.048	0.134
DistMult	0.044	0.022	0.140
ComplEx	0.043	0.028	0.152
URGE	0.598	0.576	0.730
$\texttt{UKGE}_1$	0.541	0.757	**0.810**
$\texttt{UKGE}_2$	**0.545**	**0.765**	0.794
$\texttt{UKGE}_3$	0.540	0.743	0.791

Table 6.5: Mean Spearman's Rho between predicted rankings and true rankings for relative ranking task.

metrics	CN15K		NL27k		PPI5k	
Dataset	linear	exp.	linear	exp.	linear	exp.
TransE	0.601	0.591	0.730	0.722	0.710	0.700
DistMult	0.689	0.677	0.911	0.897	0.894	0.880
ComplEx	0.723	0.712	0.921	0.913	0.896	0.881
URGE	0.302	0.300	0.267	0.265	0.726	0.723
$\texttt{UKGE}_1$	0.773	0.775	0.939	0.942	0.946	0.946
$\texttt{UKGE}_2$	0.779	0.779	0.934	0.937	0.947	0.942
$\texttt{UKGE}_3$	**0.789**	**0.790**	**0.955**	**0.956**	**0.970**	**0.969**

Table 6.6: Mean normalized DCG for global ranking task. Here *linear* stands for linear gain, and *exp.* stands for exponential gain.

NL27k NL27k is extracted from NELL (Mitchell et al., 2018), an uncertain knowledge graph obtained from webpage reading. NL27k contains 27,221 entities, 404 relations, and 175,412 uncertain relation facts. In min-max normalization, we searched for the lower boundary from 0.1 to 0.9. We found out that normalizing the confidence score to to interval $[0.1, 1]$ yields best results.

PPI5k This subset of the protein-protein interaction knowledge graph STRING contains 271,666 uncertain relation facts that consist of 5,000 proteins and 6 interactions.

Dataset	head	relation	true tail	confidence	predicted tail	predicted confidence
CN15k	rush	relatedto	fast	0.968	fast	0.703
			motion	0.709	move	0.623
	hotel	usedfor	sleeping	1.0	relaxing	0.858
			rest	0.984	sleeping	0.849
NL27k	Toyota	competeswith	Honda	0.975	Honda	0.942
			General Motors	0.930	Hyundai	0.910

Table 6.7: Examples of relation fact ranking (global) results using $\texttt{UKGE}_3$

Metrics	CN15k		NL27k		PPI5k	
Dataset	F-1	Accu.	F-1	Accu.	F-1	Accu.
TransE	0.209	69.4	0.435	50.3	0.756	83.8
DistMult	0.271	75.7	0.460	58.1	0.850	95.3
ComplEx	0.309	61.3	0.630	53.0	0.870	84.0
URGE	0.246	76.0	0.695	76.3	0.901	97.6
$\texttt{UKGE}_1$	0.467	81.0	0.760	83.0	**0.973**	**99.4**
$\texttt{UKGE}_2$	**0.479**	80.1	**0.788**	80.3	0.968	99.4
$\texttt{UKGE}_3$	0.463	**83.2**	0.778	**83.6**	0.956	99.1

Table 6.8: F-1 scores and accuracies (%) of relation fact classification

6.5.2 Model Configurations

Initialization. For objective functions $\mathcal{L}_1$and $\mathcal{L}_2$, we initialize vectors by drawing from a truncated normal distribution such that the initial $f(h, r, t)$ will not be too far from $[0, 1]$ or all fall around 0. Experiments showed our choice that drawing from a truncated normal distribution along each dimension with mean as 0, standard deviation as 0.3 perfomed well. A better initial distribution may help speed up the training convergence, but doesn't affect the final performance much. For $\mathcal{L}_3$, as the gradient of sigmoid function $\sigma(f(h, r, t))$ almost vanishes when $|f(h, r, t)| > 5$, we initialize the vectors making f(h, r, t) mostly fall around $[-5\ 5]$. Inappropriate initializer may greatly slow down the training process due to the gradient vanishing problem caused by the sigmoid function.

negative sampling. For each positive instance $(h, r, t, s) \in \mathcal{G}$, we generate n_s negative relation facts as negative samples. A recent empirical study (Kotnis and Nastase, 2017) shows that different negative sampling methods may result in different model performance. Here we conducted experiments trying both corrupting positive instances and nearest neighbor (NN) sampling. Using corrupting positive instances as the negative sampling method, as in (Bordes et al., 2013), we generate negative samples by replacing either the head or tail (positive target) with a random entity (negative target) $e \in \mathcal{E}$. In NN sampling, we choose negative targets whose embeddings are close to the positive target's embedding. This could help the model learn to discriminate between positives and negatives whose embeddings are close. In both case, we filter out samples that exist in training and validation data. Our experiments show that NN sampling achieved in higher performance and fewer epochs to converge. Therefore, we adopt NN sampling as our negative sampling method during training though its computing cost is higher.

We use the Adam optimizer for training, for which we set the exponential decay rates $\beta_1 = 0.9, \beta_2 = 0.99$. We report results for all models respectively based on their best hyperparameter settings. For each model, the setting is identified based on the validation set performance from the following sets of values: learning rate α among {0.001, 0.005, 0.01, 0.05, 0.1}, dimensionality k among {50, 75, 100, 125, 150, 200}, and batch size b among {128, 256, 512, 1024}. The L_2 regularization coefficient λ is fixed as 0.005. For $\texttt{UKGE}_1$, $\texttt{UKGE}_2$, and $\texttt{UKGE}_3$, training was stopped using early stopping based on MSE on the validation set, computed every 10 epochs. For baseline models, TransE, DistMult and ComplEx, we adopt the implementation given by (Trouillon et al., 2016), and choose the best hyper-parameters following the same grid search procedure. Training was stopped using early stopping based on MRR on the validation set. As our tasks focus on the edge relationships between nodes, we adopt the first-order proximity for the uncertain graph embedding model URGE.

6.5.3 Confidence Prediction

The objective of this task is to predict confidence scores of unknown relation facts.

Baselines. All knowledge graph embedding methods so far are aimed at deterministic knowledge graphs, and cannot predict the confidence scores. We compare our methods $\texttt{UKGE}_1$, $\texttt{UKGE}_2$, $\texttt{UKGE}_3$ to URGE which, at the best of our knowledge, is the only current uncertain graph embedding method for graphs without labeled relations.

Evaluation protocol. For each uncertain relation facts l in the test set, we predict the confidence score of l and report the mean square error (MSE) and mean absolute error (MAE). To predict confidence scores, we use Equation (6.4) for $\texttt{UKGE}_1$ and $\texttt{UKGE}_2$, and Equation (6.3) for $\texttt{UKGE}_3$ respectively.

Results. Results are reported in Table 6.3. All our models outperform the baseline URGE, since URGE only takes node proximity information, and cannot model the rich relations between entities. Our models $\texttt{UKGE}_1$, $\texttt{UKGE}_2$ perform closely, while $\texttt{UKGE}_3$ is worse than them. $\texttt{UKGE}_3$ regressed the confidence scores against plausibility values by sigmoid function. We hypothesize that this is due to that the gradient is easily vanished by the logistic function, which affects the updating of model parameters negatively.

6.5.4 Relation Fact Ranking

The next task focuses on retrieving tail entities for the query $(h, r, \underline{?t})$, and ranking these retrieved tails in the right order. We create two subtasks based on two standards: (i) *Relative ranking*, for which we rank the observed tails that fit the query. (ii) *Global ranking*, for which we rank all entities as tail candidates. The baselines we compare to include TransE (Bordes et al., 2013), DistMult (Yang et al., 2015b), and ComplEx (Trouillon et al., 2016), which have demonstrated their high performance on deterministic knowledge graphs.

6.5.4.1 Relative ranking

Given a query $(h, r, \underline{?t})$, we observe a set of tails in the uncertain knowledge graph. This task assesses if the embedding model can preserve the relative order among the observed relation facts.

Evaluation protocol. For each test query $(h, r, \underline{?t})$, we first collect from the uncertain knowledge graph all the tails that fit this query. We rank those tails t_0 according to the ground truth confidence score $s_{(h,r,t_0)}$ and our predicted score $f((h, r, t_0))$ respectively. We measure how closely the two ranking lists are correlated with Spearman's Rho (Li et al., 2009).

Results. Table 6.5 shows the mean Spearman's Rho over all test cases by different models. Note that all the previous embedding models do not incorporate the confidence score information, and they equally maximize the scores of all existing relation facts. The mean Spearman's Rho of

TransE, DistMult and ComplEx are all around 0. It is consistent with the nature of their scoring function. Our models can preserve the relative orders between observed relation facts set well. On both datasets, $\texttt{UKGE}_2$ achieves the best performance. The Spearman's Rho of $\texttt{UKGE}_2$ is around 0.545 for CN15k, and 0.767 for NL27k. $\texttt{UKGE}_2$ does not bound $f(l)$ during training, and relaxes the constraints on soft samples. This feature enables $\texttt{UKGE}_2$ to better fit the observed relation facts. This result is consistent with the previous results that $\texttt{UKGE}_2$ yields best performance in the confidence prediction task.

6.5.4.2 Global ranking

In this subtask, we rank all entities in the vocabulary as tail candidates for each test query.

Evaluation protocol. For a query $(h, r, \underline{?t})$, we rank all entities as tail candidates, and evaluate the ranking using the normalized discounted cumulative gain (nDCG). The range of nDCG is from 0 to 1. The higher nDCG, the better ranking. We define the gain in retrieving a relevant tail t_0 as the ground truth confidence score $s_{(h,r,t_0)}$. Alternatively to the linear gain, we use exponential gain to put stronger emphasis on highly relevant results. We take the mean nDCG over the test query set as our ranking metric. We report the two versions, using linear gain and exponential gain respectively, of nDCG.

Results. Table 6.6 displays the mean nDCG (using linear gain and exponential gain respectively) over all test queries for all compared methods. Though TransE, DistMult, and ComplEx do not encode the confidence score information, they maximize the plausibility of all existing relation facts and should rank the existing relation facts high. We observe that DistMult and ComplEx have considerably better performance than TransE, as TransE does not handle well `1-to-N` relations. ComplEx embeds entities and relations in complex domains and handles asymmetric relations better than DistMult. It achieves the best results among the baseline models. URGE preserve the first-order proximity Beside ranking the existing relation facts high, our model also preserve the order of the observed relation facts, and thus achieve higher nDCG scores. All the three model variants of `UKGE` outperform the baselines under all settings. Among the three variants, $\texttt{UKGE}_3$ yields best results.

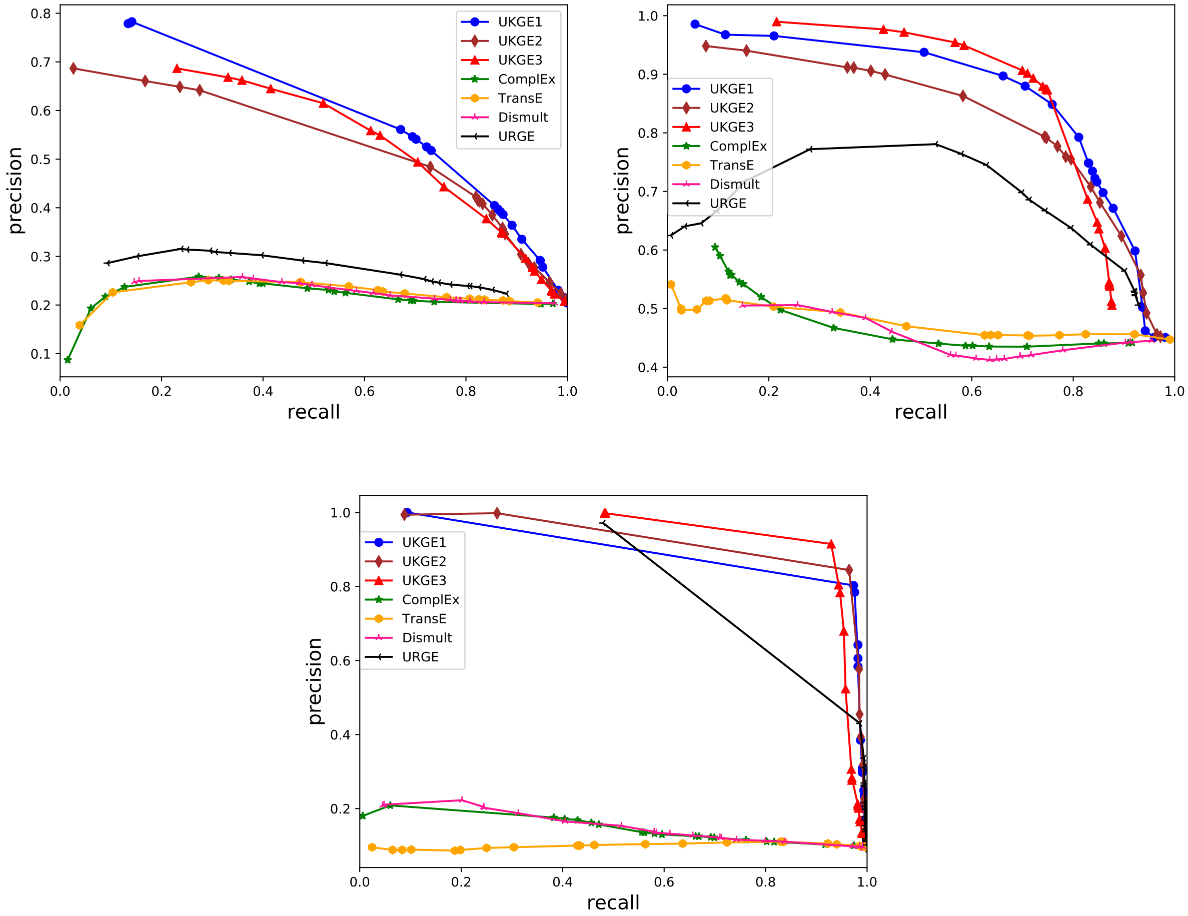


Figure 6.1: Precision-Recall curves for relation fact classification task.

Case study. Table 6.7 gives some examples of relation fact ranking results (predicting tail) by UKGE$_3$, which yields the best performance in the global ranking task. The examples are from FB This illustrates the capabilities of our model. Given a query $(h, r, \underline{?t})$, the top predicted tails and true tails are given, sorted by their score in descending order. The predictions are consistent with our common-sense.

6.5.5 Relation Fact Classification

This last task is a binary classification task to decide whether a given relation fact l is a "strong" relation fact. In an uncertain knowledge graph, a relation fact is considered strong if its confidence score s_l is above a threshold τ, otherwise weak.

Evaluation protocol. We follow a procedure that is similar to the triple classification by Wang et al. (Wang et al., 2014b). For each dataset, we set the confidence threshold τ and divide the test cases into strong relation fact group and weak relation fact group by their confidence scores. Here we set $\tau = 0.85$ for both CN15k and NL27k. Under this setting, 20.4% relation facts in ConceptNet and 20.1% in NL27k are considered strong. We fit a ID3 decision tree as a downstream classifier on the predicted confidence scores.

Results. The classification accuracy and F-1 scores are reported in table 6.8. These results show that our three models drastically outperform the best baseline model by 7.5% of accuracy and 0.17 of F-1 on CN15k, as well as by 25.5% of accuracy and 0.158 of F-1 on NL27k. Figure 1 and Figure 2 also show the Precision-Recall curves of different models, where `UKGE` variants obtain significantly higher AUPRC. Meanwhile, the difference of performance by `UKGE` variants is however marginal on this task.

6.6 Conclusion and Future Work

This chapter introduces the first work that captures uncertainty information in multi-relational representation learning. Our model `UKGE` preserves the uncertain semantic relations between entities. We propose three variants and conducted extensive experiments on relation fact confidence score prediction, relation fact ranking ranking, and classification. One meaningful future work is to substitute the plausibility scoring function with other forms, such as those with circular correlation, or bilinear mappings.

CHAPTER 7

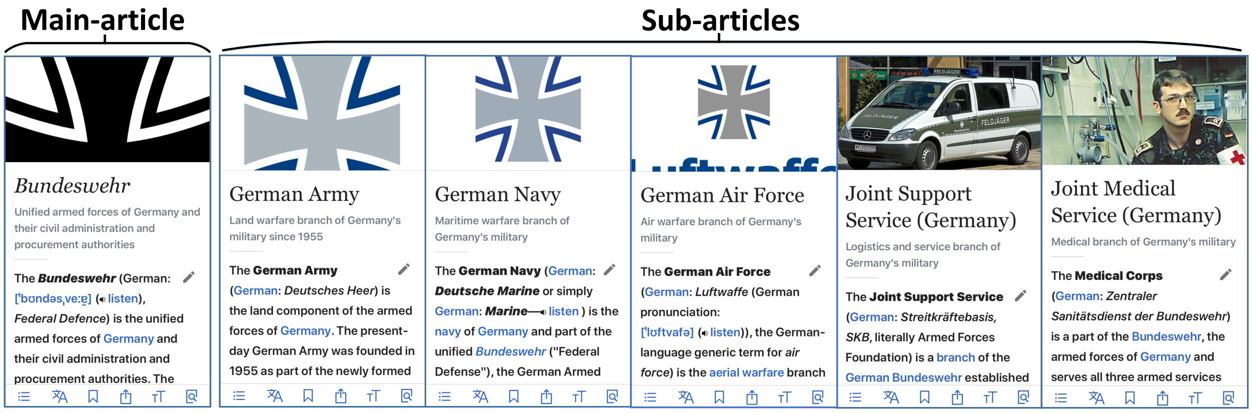
Large-scale Sub-article Relation Learning

In this chapter, we present the approach to large-scale sub-article relation learning in Wikipedia.

7.1 Introduction

Wikipedia has been the essential source of knowledge for people as well as computing research and practice. This vast storage of encyclopedia articles for real-world entities (or concepts) has brought along the automatic construction of knowledge bases (Lehmann et al., 2015; Mahdisoltani et al., 2015) that support knowledge-driven computer applications with vast structured knowledge. Meanwhile, Wikipedia also triggers the emerging of countless AI-related technologies for semantic web (Meij et al., 2014; Vrandečić and Krötzsch, 2014; Zou et al., 2014), natural language understanding (Chen et al., 2017a; Ni et al., 2016; Wang et al., 2012), content retrieval (Ackerman et al., 2013; Kittur and Kraut, 2010), and other aspects.

Most existing automated Wikipedia techniques assume the one-to-one mapping between entities and Wikipedia articles (Dojchinovski and Kliegr, 2013; Lin et al., 2017b; Mahdisoltani et al., 2015). This so-called *article-as-concept* assumption (Lin et al., 2017b) regulates each entity to be described by at most one article in a language-specific version of Wikipedia. However, recent development of Wikipedia itself is now breaking this assumption, as rich contents of an entity are likely to be separated in different articles and managed independently. For example, many details about the entity "Harry Potter" are contained in other articles such as "Harry Potter Universe", "Harry Potter influences and analogues", and "Harry Potter in translation". Such separation of



Bundeswehr

Unified armed forces of Germany and their civil administration and procurement authorities

The ***Bundeswehr*** (German: [ˈbʊndəsˌveːɐ̯] (listen), *Federal Defence*) is the unified armed forces of Germany and their civil administration and procurement authorities. The

German Army

Land warfare branch of Germany's military since 1955

The **German Army** (German: *Deutsches Heer*) is the land component of the armed forces of Germany. The present-day German Army was founded in 1955 as part of the newly formed

German Navy

Maritime warfare branch of Germany's military

The **German Navy** (German: ***Deutsche Marine*** or simply German: ***Marine***— listen) is the navy of Germany and part of the unified *Bundeswehr* ("Federal Defense"), the German Armed

German Air Force

Air warfare branch of Germany's military

The **German Air Force** (German: *Luftwaffe* (German pronunciation: [ˈlʊftvafə] (listen)), the German-language generic term for *air force*) is the aerial warfare branch

Joint Support Service (Germany)

Logistics and service branch of Germany's military

The **Joint Support Service** (German: *Streitkräftebasis, SKB*, literally Armed Forces Foundation) is a branch of the German Bundeswehr established

Joint Medical Service (Germany)

Medical branch of Germany's military

The **Medical Corps** (German: *Zentraler Sanitätsdienst der Bundeswehr*) is a part of the Bundeswehr, the armed forces of Germany and serves all three armed services

Figure 7.1: The main-article *Bundeswehr* (armed forces of Germany) and its sub-articles.

entity contents categorizes Wikipedia articles into two groups: the *main-article* that summarizes an entity, and the *sub-article* that comprehensively describes an aspect or a subtopic of the main-article. Consider another example: for the main-article *Bundeswehr* (i.e. unified armed forces of Germany) in English Wikipedia, we can find its split-off sub-articles such as *German Army*, *German Navy*, *German Airforce*, *Joint Support Service of Germany*, and *Joint Medical Service of Germany* (as shown in Fig. 7.1). This type of sub-article splitting is quite common on Wikipedia. Around 71% of the most-viewed Wikipedia entities are split-off to an average of 7.5 sub-articles (Lin et al., 2017b).

While sub-articles may enhance human readabilities, the violation of the article-as-concept assumption is problematic to a wide range of Wikipedia-based technologies and applications that critically rely on this assumption. For instance, Wikipedia-based knowledge base construction (Lehmann et al., 2015; Mahdisoltani et al., 2015; Mousavi et al., 2014a) assumes the article title as an entity name, which is then associated with the majority of relation facts for the entity from the infobox of the corresponding article. Split-off of articles sow confusion in a knowledge base extracted with these techniques. Clearly, explicit (Gabrilovich and Markovitch, 2007; Hecht et al., 2012) and implicit semantic representation techniques (Chen and Zaniolo, 2017; Schuhmacher and Ponzetto, 2014; Xie et al., 2016) based on Wikipedia are impaired due to that a part of links and text features utilized by these approaches are now likely to be isolated from the entities, which further affects NLP tasks based on these semantic representations such as semantic relatedness analysis (Liu et al., 2016; Ni et al., 2016; Strube and Ponzetto, 2006), relation extraction (Chen et al., 2018f; Mousavi et al., 2014a), and named entity disambiguation (Yamada et al., 2016). Multilingual tasks such as knowledge alignment (Chen et al., 2017c, 2018e; Suchanek et al., 2011; Wang et al., 2012) and cross-lingual Wikification (Tsai and Roth, 2016) become challenging for entities with multiple articles, since these tasks assume that we have a one-to-one match between articles in both languages. Semantic search (Cai et al., 2013; Meij et al., 2014; Zou et al., 2014) is also affected due to the diffused nature of the associations between entities and articles.

To support the above techniques and applications, it is vital to address the *sub-article matching* problem, which aims to restore the complete view of each entity by matching the sub-articles back to the main-article. However, it is non-trivial to develop a model which recognizes the implicit relations that exist between main and sub-articles. A recent work (Lin et al., 2017b) has attempted to tackle this problem by characterizing the match of main and sub articles using some explicit features. These features focus on measuring the symbolic similarities of article and section titles, the structural similarity of entity templates and page links, as well as cross-lingual co-occurrence. Although these features are helpful to identify the sub-article relations among a small scale of article pairs, they are still far from fully characterizing the sub-article relations in the large body of Wikipedia. And more importantly, the semantic information contained in the titles and text contents, which is critical to the characterization of semantic relations of articles, has not been used for this task. However, effective utilization of the semantic information would also require a large collection of labeled main and sub-article pairs to generalize the modeling of such implicit features.

In this chapter, we introduce a new approach for addressing the sub-article matching problem. Our approach adopts neural document encoders to capture the semantic features from the titles and text contents of a candidate article pair, for which several encoding techniques are explored with. Besides the semantic features, the model also utilizes a set of explicit features that measure the symbolic and structural aspects of an article pair. Using a combination of these features, the model decides whether an article is the sub-article of another. To generalize the problem, massive crowdsourcing and strategic rules are applied to create a large dataset that contains around 196k Wikipedia article pairs, where around 10% are positive matches of main and sub-articles, and the rest comprehensively cover different patterns of negative matches. Held-out estimation proves effectiveness of our approach by significantly outperforming previous baselines, and reaching near-perfect precision and recall for detecting positive main and sub-article matches from all candidates. To show the practicability of our approach, we also employ our model to extract main and sub-article matches in the entire English Wikipedia using a 3,000-machine MapReduce (Dean and Ghemawat, 2008). This process has produced a large collection of new high-quality main and sub-article matches, and are being migrated into a production knowledge base.

7.2 Related Work

A recent work (Lin et al., 2017b) has launched the first attempt to address the Wikipedia sub-article matching problem, in which the authors have defined the problem into the binary classification of candidate article pairs. Each article pair is characterized based on a group of explicit features that lies in three categories: (1) symbolic similarity: this includes token overlapping of the titles, maximum token overlapping among the section titles of the two articles, and term frequencies of the main-article titles in the candidate sub-article contents; (2) structural similarity: this includes structure similarity of article templates, link-based centrality measures and the Milne-Witten Index (Milne and Witten, 2008); (3) cross-lingual co-occurrence: these features consider the proportion of languages where the given article pairs have been identified as main and sub-articles, and the relative multilingual "globalness" measure of the candidate main-article. Although some statistical classification models learnt on these explicit features have offered satisfactory accuracy of binary classification on a small dataset of 3k article pairs that cover a subset of the most-viewed Wikipedia articles, such simple characterization is no-doubt far from generalizing the problem. When the dataset scales up to the range of the entire Wikipedia, it is very easy to find numerous counterfactual cases for these features. Moreover, the cross-lingual co-occurrence-based features are not generally usable due to the incompleteness of inter-lingual links that match the cross-lingual counterparts of Wikipedia articles. Some recent works have even pointed out that such cross-lingual alignment information only covers less than 15% of the articles (Chen et al., 2017c; Lehmann et al., 2015; Vrandečić, 2012). More importantly, we argue that the latent semantic information of the articles should be captured, so as to provide more generalized and comprehensive characterization of the article relation.

Sentence or article matching tasks such as textual entailment recognition (Hu et al., 2014; Sha et al., 2016; Yin et al., 2016a) and paraphrase identification (Yin and Schütze, 2015; Yin et al., 2016a) require the model to identify content-based discourse relations of sentences or paragraphs (Lascarides and Asher, 1993), which reflect logical orders and semantic consistency. Many recent efforts adopt different forms of deep neural document encoders to tackle these tasks, where several encoding techniques have been widely employed, including convolutional neural networks (Hu et al., 2014; Yin and Schütze, 2015), recurrent neural networks (Sha et al., 2016), and attentive techniques (Kadlec et al., 2016; Rocktäschel et al., 2016a; Yin et al., 2016a). Detection of the sub-article relations requires the model to capture a high-level understanding of both contents and text structuring of articles. Unlike the previously mentioned discourse relations, the sub-article relations can be reflected from different components of Wikipedia articles including titles, text contents and link structures. To tackle new and challenging task of sub-article matching, we incorporate neural document encoders with explicit features in our model, so as to capture the sub-article relation based on both symbolic and semantic aspects of the Wikipedia article pairs. Meanwhile, we also take efforts to prepare a large collection of article pairs that seek to well generalize the problem.

7.3 Modeling

In this section, we introduce the proposed model for the Wikipedia sub-article matching task. We begin with the denotations and problem definition.

7.3.1 Preliminaries

Denotations. We use W to denote the set of Wikipedia articles, in which we model an article $A_i \in W$ as a triple $A_i = (t_i, c_i, s_i)$. t_i is the title, c_i the text contents, and s_i the miscellaneous structural information such as templates, sections and links. $t_i = \{w_{t1}, w_{t2}, ..., w_{tl}\}$ and $c_i = \{w_{c1}, w_{c2}, ..., w_{cm}\}$ thereof are both sequences of words. In practice, we use the first paragraph of A_i to represent c_i since it is the summary of the article contents. For each word w_i, we use bold-faced $\mathbf{w}_i$ to denote its embedding representation. We use $F(A_i, A_j)$ to denote a sequence of explicit features that provide some symbolic and structural measures for titles, text contents and link structures, which we are going to specify in Section 7.3.3. We assume that all articles are written in the same language, as it is normally the case of main-articles and their sub-articles. In this chapter, we only consider English articles *w.l.o.g*.

Problem definition. *Sub-article matching* is defined as a binary classification problem on a set of candidate article pairs $P \subseteq W \times W$. Given a pair of articles $p = (A_i, A_j) \in P$, a model should decide whether A_j is the sub-article of A_i. The problem definition complies with the previous work that first introduces the problem (Lin et al., 2017b), and is related to other sentence matching problems for discourse relations such as text entailment and paraphrase identification (Poria et al., 2015; Sha et al., 2016; Yin and Schütze, 2015).

The sub-article relation is qualified based on two criteria, i.e. A_j is a sub-article of A_i if (1) A_j describes an aspect or a subtopic of A_i, and (2) c_j can be inserted as a section of A_i without breaking the topic summarized by t_i. It is noteworthy that the sub-article relation is anti-symmetric, i.e. if A_j is a sub-article of A_i then A_i is not a sub-article of A_j. We follow these two criteria in the crowdsourcing process for dataset creation, as we are going to explain in Section 7.4. To address the sub-article matching problem, our model learns on a combination of two aspects of the Wikipedia articles. Neural document encoders extract the implicit semantic features from text, while a series of explicit features are incorporated to characterize the symbolic or structural aspects. In the following, we introduce each component of our model in detail.

7.3.2 Document Encoders

A neural document encoder $E(X)$ encodes a sequence of words X into a latent vector representation of the sequence. We investigate three widely-used techniques for document encoding (Hu et al., 2014; Rocktäschel et al., 2016a; Sha et al., 2016; Yin and Schütze, 2015; Yin et al., 2016a), which lead to three types of encoders for both titles and text contents of Wikipedia articles, i.e. convolutional encoders (CNN), gated recurrent unit encoders (GRU), and attentive encoders. Defition of these neural sequence encoding techniques has been presented in Section 2.3.

We adopt each one of the three encoding techniques, i.e. the convolution layer with pooling, the GRU, and the attentive GRU, to form three types of document encoders respectively. Each encoder consists of one or a stack of the corresponding layers depending on the type of the input document, and encodes the document into a embedding vector.

7.3.3 Explicit Features

In addition to the implicit semantic features provided by document encoders, we define explicit features $F(A_i, A_j) = \{r_{tto}, r_{st}, r_{indeg}, r_{mt}, f_{TF}, I_{MW}, r_{outdeg}, d_{te}, r_{dt}\}$. A portion of the explicit features are carried forward from (Lin et al., 2017b) to provide some token-level and structural measures of an article pair (A_i, A_j):

- r_{tto}: token overlap ratio of titles, i.e. the number of overlapped words between t_i and t_j divided by $|t_i|$.
- r_{st}: the maximum of the token overlap ratios among the section titles of A_i and those of A_j.
- r_{indeg}: the in-degree ratio, which is the number of incoming links in A_i divided by that of A_j. r_{indeg} measures the relative centrality of A_i with regard to A_j.
- r_{mt}: the maximum of the token overlap ratios between any anchor title of the main-article template [1] of A_i and t_j, or zero if the main-article template does not apply to A_i.
- f_{TF}: normalized term frequency of t_i in c_j.
- d_{MW}: Milne-Witten Index (Milne and Witten, 2008) of A_i and A_j, which measures the similarity of incoming links of two articles via the Normalized Google Distance (Cilibrasi and Vitanyi, 2007).

In addition to the above features, we also include the following features.

[1] https://en.wikipedia.org/wiki/Template:Main

- r_{outdeg}: the out-degree ratio, which measures the relative centrality of A_i and A_j similar to r_{indeg}.

- d_{te}: the average embedding distance of tokens in titles t_i and t_j.

- r_{dt}: token overlap ratio of c_i and c_j, which is used in (Lin and Hovy, 2002) to measure the document relatedness. The calculation of r_{dt} is based on the first paragraphs.

We normalize the distance and frequency-based features (i.e. f_{TF} and d_{te}. d_{MW} is already normalized by its definition.) using min-max rescaling. Note that, we do not preserve the two cross-lingual features in (Lin et al., 2017b). This is because, in general, these two cross-lingual features are not applicable when the candidate space scales up to the range of the entire Wikipedia, since the inter-lingual links that match articles across different languages are far from complete (Chen et al., 2017c; Lehmann et al., 2015).

7.3.4 Training

Learning objective. The overall architecture of our model is shown in Fig. 7.2. The model characterizes each given article pair $p = (A_i, A_j) \in P$ in two stages.

1. Four document encoders (of the same type) are used to encode the titles and text contents of A_i and A_j respectively, which are denoted as $E_t^{(1)}(t_i)$, $E_t^{(2)}(t_j)$, $E_c^{(1)}(c_i)$ and $E_c^{(2)}(c_j)$. Two logistic regressors realized by sigmoid multi-layer perceptrons (MLP) (Bengio, 2009) are applied on $E_t^{(1)}(t_i) \oplus E_t^{(2)}(t_j)$ and $E_c^{(1)}(c_i) \oplus E_c^{(2)}(c_j)$ to produce two confidence scores s_t and s_c for supporting A_j to be the sub-article of A_i.

2. The two semantic-based confidence scores are then concatenated with the explicit features ($\{s_t, s_c\} \oplus F(A_i, A_j)$), to which another linear MLP is applied to obtain the two confidence scores $\hat{s}_p^+$ and $\hat{s}_p^-$ for the boolean labels of positive prediction l^+ and negative prediction l^- respectively. Finally, $\hat{s}_p^+$ and $\hat{s}_p^-$ are normalized by binary softmax functions $s_p^+ = \frac{\exp(\hat{s}_p^+)}{\exp(\hat{s}_p^+)+\exp(\hat{s}_p^-)}$ and $s_p^- = \frac{\exp(\hat{s}_p^-)}{\exp(\hat{s}_p^+)+\exp(\hat{s}_p^-)}$.

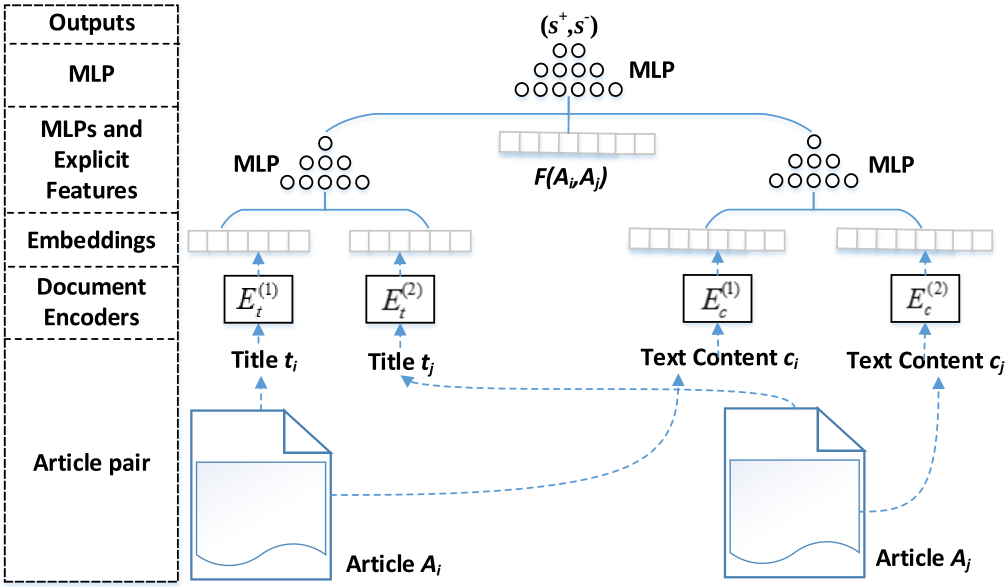


Figure 7.2: Learning architecture of the model.

The learning objective is to minimize the following binary cross-entropy loss.

$$L = -\frac{1}{|P|}\sum_{p\in P}\left(l^+ \log s_p^+ + l^- \log s_p^-\right)$$

Annotated word embeddings. We pre-train the Skipgram (Mikolov et al., 2013c) word embeddings on the English Wikipedia dump to support the input of the article titles and text contents to the model, as well as the calculation of the feature d_{te}. We parse all the inline hyperlinks of Wikipedia dump to the corresponding article titles, and tokenize the article titles in the plain text corpora via Trie-based maximum token matching. This tokenization process aims at including Wikipedia titles in the vocabulary of word embeddings. Although this does not ensure all the titles to be involved, as some of them occur too rarely in the corpora to meet the minimum frequency requirement of the word embedding model. This tokenization process is also adopted during the calculation of d_{te}. After pre-training, we fix the word embeddings to convert each document to a sequence of vectors to be fed into the document encoder.

7.4 Experiments

In this section, we present the experimental evaluation of the proposed model. We first create a dataset that contains a large collection of candidate article pairs for the sub-article matching problem. Then we compare variants of the proposed model and previous approaches based on held-out estimation on the dataset. Lastly, to show the practicability of our approach, we train the model on the full dataset, and perform predictions using MapReduce on over 108 million candidate article pairs extracted from the entire English Wikipedia.

Dataset Preparation. We have prepared a new dataset, denoted WAP196k, in two stages. We start with producing positive cases via crowdsourcing. In detail, we first select a set of articles, where each article title concatenates two Wikipedia entity names directly or with a proposition, e.g. *German Army* and *Fictional Universe of Harry Potter*. We hypothesize that such articles are more likely to be a sub-article of another Wikipedia article. Note that this set of articles exclude the pages that belong to a meta-article category such as *Lists* [2] and *Disambiguation* [3], which usually do not have text contents. Then we sample from this set of articles for annotation in the internal crowdsourcing platform of Google. For each sampled article, we follow the criteria in Section 7.3.1 to instruct the annotators to decide whether it is a sub-article, and to provide the URL to the corresponding main-article if so. Each crowdsourced article has been reviewed by three annotators, and is adopted for the later population process of the dataset if total agreement is reached. Within three months, we have obtained 17,349 positive matches of main and sub-article pairs for 5,012 main-articles, and around 4k other negative identifications of sub-articles.

Based on the results of crowdsourcing, we then follow several strategies to create negative cases: (1) For each positive match (A_i, A_j), we insert the inverted pair (A_j, A_i) as a negative case based on the anti-symmetry of sub-article relations, therefore producing 17k negative cases; (2) For each identified main-article, if multiple positively matched sub-articles coexist, such sub-articles are paired into negative cases as they are considered as "same-level articles". This step contributes around 27k negative cases; (3) We substitute A_i with other articles that are pointed by an inline hyperlink in c_j, or substitute A_j with samples from the 4k negative identifications of sub-articles

[2] `https://en.wikipedia.org/wiki/Category:Lists`

[3] `https://en.wikipedia.org/wiki/Wikipedia:Disambiguation`

in stage 1. We select a portion from this large set of negative cases to ensure that each identified main-article has been paired with at least 15 negative matches of sub-articles. This step contributes the majority of negative cases. In the dataset, we also discover that around 20k negative cases are measured highly (> 0.6) by at least one of the symbolic similarity measures r_{tto}, r_{st} or f_{TF}.

The three strategies for negative case creation seek to populate the WAP196k dataset with a large amount of negative matches of articles that represent different counterfactual cases. The statistics of WAP196k are summarized in Table 7.1, which indicate it to be much more large-scale than the dataset used by previous approaches (Lin et al., 2017b) that contains around 3k article pairs. The creation of more negative cases than positive cases is in accord with the general circumstances of the Wikipedia where the sub-article relations hold for a small portion of the article pairs. Hence, the effectiveness of the model should be accordingly evaluated by how precisely and completely it can recognize the positive matches from all candidate pairs from the dataset. As we have stated in Section 7.3.1, we encode the first paragraph of each article to represent its text contents.

7.4.1 Evaluation

We use a held-out estimation method to evaluate our approach on WAP196k. Besides three proposed model variants that combine a specfic type of neural document encoders with the explicit features, we compare several statistical classification algorithms that (Lin et al., 2017b) have trained on the explicit features. We also compare with three neural document pair encoders without explicit features that represent the other line of related work (Hu et al., 2014; Rocktäschel et al., 2016a; Sha et al., 2016; Yin and Schütze, 2015).

Model configurations. We use AdaGrad to optimize the learning objective function and set the learning rate as 0.01, batchsize as 128. For document encoders, we use two convolution/GRU/attentive GRU layers for titles, and two layers for the text contents. When inputting articles to the document encoders, we remove stop words in the text contents, zero-pad short ones and truncate overlength ones to the sequence length of 100. We also zero-pad short titles to the sequence length of 14, which is the maximum length of the original titles. The dimensionality of document em-

Table 7.1: Statistics of the dataset.

#Article pairs	#Positive cases	#Negative cases	#Main-articles	#Distinct articles
195,960	17,349	178,611	5,012	32,487

Table 7.2: Cross-validation results on WAP196k. We report precision, recall and F1-scores on three groups of models: (1) statistical classification algorithms based on explicit features, including logistic regression, Naive Bayes Classifier (NBC), Linear SVM, Adaboost (SAMME.R algorithm), Decision Tree (DT), Random Forest (RF) and k-nearest-neighbor classifier (kNN); (2) three types of document pair encoders without explicit features; (3) the proposed model in this chapter that combines explicit features with convolutional document pair encoders (CNN+F), GRU encoders (GRU+F) or attentive encoders (AGRU+F).

Model	Explicit Features						
	Logistic	NBC	Adaboost	LinearSVM	DT	RF	kNN
Precision (%)	82.64	61.78	87.14	82.79	87.17	89.22	65.80
Recall (%)	88.41	87.75	85.40	89.56	84.53	84.49	78.66
F1-score	0.854	0.680	0.863	0.860	0.858	0.868	0.717

Model	Semantic Features			Model	Explicit+Semantic		
	CNN	GRU	AGRU		CNN+F	GRU+F	AGRU+F
Precision (%)	95.83	95.76	93.98	Precision (%)	**99.13**	98.60	97.58
Recall (%)	90.46	87.24	86.47	Recall (%)	**98.06**	88.47	86.80
F1-score	0.931	0.913	0.901	F1-score	**0.986**	0.926	0.919

beddings is selected among {100, 150, 200, 300}, for which we fix 100 for titles and 200 for text contents. For convolutional encoders, we select the kernel size and the pool size from 2 to 4, with the kernel size of 3 and 2-max-pooling adopted. For pre-trained word embeddings, we use context size of 20, minimum word frequency of 7 and negative sampling size of 5 to obtain 120-dimensional embeddings from the tokenized Wikipedia corpora mentioned in Section 7.3.4. Following the convention, we use one hidden layer in MLPs, where the hidden size averages those of the input and output layers.

Evaluation protocal. Following (Lin et al., 2017b), we adopt 10-fold cross-validation in the evaluation process. At each fold, all models are trained till converge. We aggregate *precision*, *recall* and *F1-score* on the positive cases at each fold of testing, since the objective of the task is to effectively identify the relatively rare article relation among a large number of article pairs. All three metrics are preferred to be higher to indicate better performance.

Results. Results are reported in Table 7.2. The explicit features alone are helpful to the task,

Table 7.3: Ablation on feature categories for CNN+F.

Features	Precision	Recall	F1-score
All features	99.13	98.06	0.986
No titles	98.03	85.96	0.916
No text contents	98.55	95.78	0.972
No explicit	95.83	90.46	0.931
Explicit only	82.64	88.41	0.854

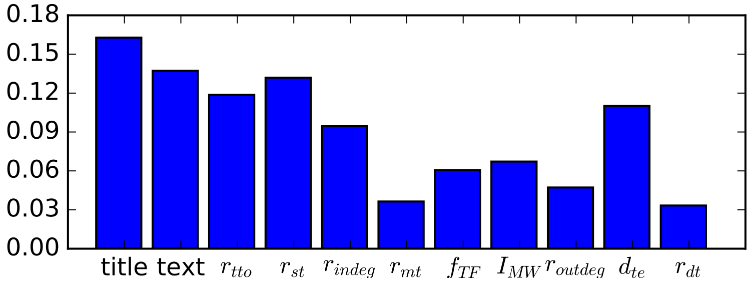


Figure 7.3: Relative importance (RI) of features analyzed by Garson's algorithm. RI of each feature is aggregated from all folds of cross-validation.

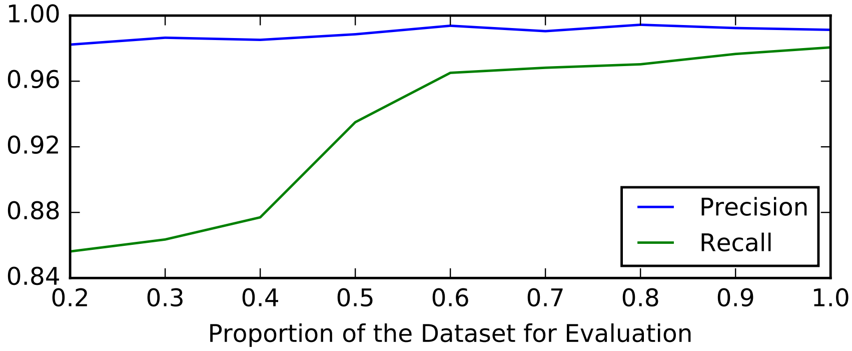


Figure 7.4: Sensitivity of CNN+F on the proportion of dataset for evaluation.

on which the result by the best baseline (Random Forest) is satisfactory. However, the neural encoders for document pairs, even without the explicit features, outperform Random Forest by 4.76% on precision, 1.98% on recall and 0.033 on F1-score. This indicates that the implicit semantic features are critical for characterizing the matching of main and sub-articles. Among the three types of document encoders, the convolutional encoder is more competent than the rest two sequence encoders, which outperforms Random Forest by 6.54% of precision, 8.97% of recall and 0.063 of F1-score. This indicates that the convolutional and pooling layers that effectively capture the local semantic features are key to the identification of sub-article relations, while such relations appear to be relatively less determined by the sequence information and overall document meanings that are leveraged by the GRU and attention encoders. The results by the proposed model which combines document pair encoders and explicit features are very promising. Among these, the model variant with convolutional encoders (CNN+F) obtained close to perfect precision and recall.

Meanwhile, we perform ablation on different categories of features and each specific feature, so as to understand their significance to the task. Table 7.3 presents the ablation of feature categories for the CNN-based model. We have already shown that completely removing the implicit semantic features would noticeably impair the precision. Removing the explicit features moderately hinders both precision and recall. As for the two categories of semantic features, we find that removing either of them would noticeably impair the model performance in terms of recall, though the removal of title embeddings has much more impact than that of text content embeddings. Next, we perform Garson's algorithm (Féraud and Clérot, 2002; Olden and Jackson, 2002) on the weights of the last linear MLP of CNN+F to analyze the relative importance (RI) of each specific feature, which are reported as Fig. 7.3. It is noteworthy that, besides the text features, the explicit features r_{tto}, r_{st} and d_{te} that are related to article or section titles also show high RI. This is also close to the practice of human cognition, as we humans are more likely to be able to determine the semantic relation of a given pair of articles based on the semantic relation of the titles and section titles than based on other aspects of the explicit features.

Furthermore, to examine how much our approach may rely on the large dataset to obtain a generic solution, we conduct a sensitivity analysis of CNN+F on the proportion of the dataset used for cross-validation, which is reported in Fig. 7.4. We discover that, training the model on smaller portions of the dataset would decrease the recall of predictions by the model, though the impact on the precision is very limited. However, even using 20% of the data, CNN+F still obtains better precision and recall than the best baseline Random Forest that is trained solely on explicit features in the setting of full dataset.

To summarize, the held-out estimation on the WAP196k dataset shows that the proposed model is very promising in addressing the sub-article matching task. Considering the large size and heterogeneity of the dataset, we believe the best model variant CNN+F is close to a well-generalized solution.

Table 7.4: Examples of recognized main and sub-article matches. The italicize sub-article titles are without overlapping tokens with the main article titles.

Main Article	Sub-articles
Outline of government	*Bicameralism*, *Capitalism*, *Dictatorship*, *Confederation*, *Oligarchy*, *Sovereign state*
Computer	Computer for operations with functions, Glossary of computer hardware terms, Computer user, *Timeline of numerical analysis after 1945*, Stored-program computer, Ternary computer
Hebrew alphabet	Romanization of Hebrew
Recycling by material	Drug recycling, *Copper*, *Aluminium*
Chinese Americans	History of Chinese Americans in Dallas-Fort Worth, History of Chinese Americans in San Francisco, Anti-Chinese Violence in Washington
Genetics	Modification (Genetics), Theoretical and Applied Genetics, Encyclopedia of Genetics
Service Rifle	United States Marine Corps Squad Advanced Marksman Rifle, United States Army Squad Designated Marksman Rifle
Transgender rights	LGBT rights in Panama, LGBT rights in the United Arab Emirates, Transgender rights in Argentina, History of transgender people in the United States
Spectrin	Spectrin Repeat
Geography	Political Geography, Urban geography, Visual geography, *Colorado Model Content Standards*
Nuclear Explosion	Outline of Nuclear Technology, International Day Against Nuclear Tests
Gay	*LGBT Rights by Country or Territory*, Philadelphia Gay News, Troll (gay slang), Gay literature
FIBA Hall of Fame	*Šarūnas Marčiulionis*
Arve Isdal	*March of the Norse*, *Between Two Worlds*
Independent politician	*Balasore (Odisha Vidhan Sabha Constituency)*
Mathematics	Hierarchy (mathematics), *Principle part*, Mathematics and Mechanics of Complex Systems, *Nemytskii operator*, *Spinors in three dimensions*, *Continuous functional calculus*, Quadrature, Table of mathematical symbols by introduction date, *Hasse invariant of an algebra*, Concrete Mathematics
Homosexuality	*LGBT rights in Luxembourg*, List of Christian denominational positions on homosexuality
Bishop	*Roman Catholic Diocese of Purnea*, *Roman Catholic Diocese of Luoyang*
Lie algebra	Radical of a Lie algebra, Restricted Lie algebra, *Adjoint representation*, Lie Group

7.4.2 Mining Sub-articles from the Entire English Wikipedia

For the next step, we move on to putting the proposed model into production by serving it to identify the main and sub-article matching on the entire body of the English Wikipedia. The English Wikipedia contains over 5 million articles, which lead to over 24 trillion ordered article pairs. Hence, instead of serving our model on that astronomical candidate space, we simplify the task by predicting only for each article pair that forms an inline hyperlink across Wikipedia pages, except for those that appear already in the *main-article* templates. This reduces our candidate space to about 108 million article pairs.

We train the best model variant CNN+F from the previous experiment for serving. We carry forward the model configurations from the previous experiment. The model is trained on the entire

WAP196k dataset till converge. The extraction of the candidate article pairs as well as the serving of the model is conducted via MapReduce on 3,000 machines, which lasts around 9 hours in total. We select the 200,000 positive predictions with highest confidence scores s_p^+, based on which human evaluation on three turns of 1,000 sampled results estimates a 85.7$\%$ of $P@200k$ (precision at top 200,000 predictions). Examples of identified main and sub-article matches are listed in Table 7.4. Based on the selected positive predictions, the number of sub-articles per main-article is estimated as 4.9, which is lower than 7.5 that is estimated on the 1,000 most viewed articles by (Lin et al., 2017b). There are also around 8$\%$ of sub-articles that are paired with more than one main-articles. Based on the promising results from the large-scale model serving, our team is currently working on populating the identified sub-article relations into the backend knowledge base for our search engine.

7.5 Conclusion and Future Work

In this section, we have proposed a neural article pair model to address the sub-article matching problem in Wikipedia. The proposed model utilizes neural document encoders for titles and text contents to capture the latent semantic features from Wikipedia articles, for which three types of document encoders have been considered, including the convolutional, GRU and attentive encoders. A set of explicit features are incorporated into the learning framework that comprehensively measured the symbolic and structural similarity of article pairs. We have created a large article pair dataset WAP196k from English Wikipedia which seeks to generalize the problem with various patterns of training cases. The experimental evaluation on WAP196k based on cross-validation shows that the document encoders alone are able to outperform the previous models using only explicit features, while the combined model based on both implicit and explicit features is able to achieve near-perfect precision and recall. Large-scale serving conducted on the entire English Wikipedia is able to produce a large amount of new main and sub-article matches with promising quality. For future work, it is natural to apply the proposed model to other language-versions of Wikipedia for production. It is also meaningful to develop an approach to differentiate the sub-articles that describe refined entities and those that describe abstract sub-concepts.

CHAPTER 8

Multifaceted Protein-Protein Interaction Learning

In this section, we present the state-of-the-art method for learning protein-protein interactions based on raw amino acid sequences (Chen et al., 2019b). This work seeks provide an approach to populate multi-relational data in protein knowledge bases (Moal and Fernández-Recio, 2012; Szklarczyk et al., 2016).

8.1 Introduction

Detecting protein-protein interactions (PPIs) and characterizing the interaction types are essential toward understanding cellular biological processes in normal and disease states. Knowledge from these studies potentially facilitates therapeutic target identification (Petta et al., 2016) and novel drug design (Skrabanek et al., 2008). High-throughput experimental technologies have been rapidly developed to discover and validate PPIs on a large scale. These technologies include yeast two-hybrid screens (Fields and Song, 1989), tandem affinity purification (Gavin et al., 2002), and mass spectrometric protein complex identification (Ho et al., 2002). However, experiment-based methods remain expensive, labor-intensive, and time-consuming. Most importantly, they often suffer from high levels of false-positive predictions (Sun et al., 2017a; You et al., 2015). Evidently, there is an immense need for reliable computational approaches to identify and characterize PPIs.

The amino acid sequence represents the primary structure of a protein, which is the simplest type of information either obtained through direct sequencing or translated from DNA sequences. Many research efforts address the PPI problem based on predefined features extracted from protein sequences, such as ontological features of amino acids (Jansen et al., 2003), autocovariance (AC) (Guo et al., 2008), conjoint triads (CT) (Shen et al., 2007) and composition-transition-distribution (CTD) descriptors (Yang et al., 2010). These features generally summarize specific aspects of protein sequences such as physicochemical properties, frequencies of local patterns, and the positional distribution of amino acids. On top of these features, several statistical learning algorithms (Guo et al., 2008; Huang et al., 2015; You et al., 2015, 2014) are applied to predict PPIs in the form of binary classification. These approaches provide feasible solutions to the problem. However, the extracted features used in these approaches only have limited coverage on interaction information, as they are dedicated to specific facets of the protein profiles.

To mitigate the inadequacy of statistical learning methods, deep learning algorithms provide the powerful functionality to process large-scale data and automatically extract useful features for objective tasks (LeCun et al., 2015). Recently, deep learning architectures have produced powerful systems to address several bioinformatics problems related to single nucleotide sequences, such as genetic variants detection (Anderson, 2018), DNA function classification (Quang and Xie, 2016), RNA-binding site prediction (Zhang et al., 2015) and chromatin accessibility prediction (Min et al., 2017). These works typically use convolutional neural networks (CNN) (Anderson, 2018; Zhang et al., 2015) for automatically selecting local features, or recurrent neural networks (RNN) (Quang and Xie, 2016) that aim at preserving the contextualized and long-term ordering information. By contrast, fewer efforts (discussed in Related Work) have been made to capture the pairwise interactions of proteins with deep learning, which remains a non-trivial problem with the following challenges: (i) Characterization of the proteins requires a model to effectively filter and aggregate their local features, while preserving significant contextualized and sequential information of the amino acids; (ii) Extending a deep neural architecture often leads to inefficient learning processes, and suffers from the notorious vanishing gradient problem (Pascanu et al., 2013); (iii) An effective mechanism is also needed to apprehend the mutual influence of protein pairs in PPI prediction. Moreover, it is essential for the framework to be scalable to large data, and to be generalized to different prediction tasks.

In this chapter, we introduce `PIPR` (Protein-Protein Interaction Prediction Based on Siamese

Residual RCNN), a deep learning framework for PPI prediction using only the sequences of a protein pair. PIPR employs a Siamese architecture to capture the mutual influence of a protein sequence pair. The learning architecture is based on a residual recurrent convolutional neural network (RCNN), which integrates multiple occurrences of convolution layers and residual gated recurrent units. To represent each amino acid in this architecture, PIPR applies an efficient property-aware lexicon embedding approach to better capture the contextual and physicochemical relatedness of amino acids. This comprehensive encoding architecture provides a multi-granular feature aggregation process to effectively leverage both sequential and robust local information of the protein sequences. It is important to note that the scope of this work focuses only on the primary sequence as it is the fundamental information to describe a protein.

Our contributions are three-fold. First, we construct an end-to-end framework for PPI prediction that relieves the data pre-processing efforts for users. PIPR requires only the primary protein sequences as the input, and is trained to automatically preserve the critical features from the sequences. Second, we emphasize and demonstrate the needs of considering the contextualized and sequential information when modeling the PPIs. Third, the architecture of PIPR can be flexibly used to address different PPI tasks. Besides the binary prediction that is widely attempted in previous works, our framework extends its use to two additional challenging problems: multi-class interaction type prediction and binding affinity estimation. We use five datasets to evaluate the performance of our framework on these tasks. PIPR outperforms various state-of-the-art approaches on the binary prediction task, which confirms the effectiveness in terms of integrating both local features and sequential information. The promising performance of the other two tasks demonstrates the wide usability of our approach. Especially on the binding affinity estimation of mutated proteins, PIPR is able to respond to the subtle changes of point mutations and provides the best estimation with the smallest errors.

8.2 Related Work

Sequence-based approaches provide a critical solution to the binary PPI prediction task. Homology-based methods (Philipp et al., 2016) rely on BLAST to map a pair of sequences to known interacting proteins. Alternatively, other works address the task with statistical learning models, including SVM (Guo et al., 2008; You et al., 2014), kNN (Yang et al., 2010), Random Forest (Wong et al., 2015), multi-layer perceptron (MLP) (Du et al., 2017), and ensemble ELM (EELM) (You et al., 2013). These approaches rely on several feature extraction processes for the protein sequences, such as CT (Sun et al., 2017a; You et al., 2013), AC (Guo et al., 2008; Sun et al., 2017a; You et al., 2013), CTD (Du et al., 2017; Yang et al., 2010), multi-scale continuous and discontinuous (MCD) descriptors (You et al., 2013), and local phase quantization (LPQ) (Wong et al., 2015). These features measure physicochemical properties of the 20 canonical amino acids, and aim at summarizing full sequence information relevant to PPIs. More recent works (Sun et al., 2017a; Wang et al., 2017b) propose the use of stacked autoencoders (SAE) to refine these heterogeneous features in low-dimensional spaces, which improve the aforementioned models on the binary prediction task. On the contrary, fewer efforts have been made towards multi-class prediction to infer the interaction types (Silberberg et al., 2014; Zhu et al., 2006) and the regression task to estimate binding affinity (Srinivasulu et al., 2015; Yugandhar and Gromiha, 2014). These methods have largely relied on their capability of extracting and selecting better features, while the extracted features are far from fully exploiting the interaction information.

By nature, the PPI prediction task is comparable to the neural sentence pair modeling tasks in natural language processing (NLP) research, as they both seek to characterize the mutual influence of two sequences based on their latent features. In NLP, neural sentence pair models typically focus on capturing the discourse relations of lexicon sequences, such as textual entailment (Hu et al., 2014; Yin et al., 2016a), paraphrases (He et al., 2015; Yin and Schütze, 2015) and sub-topic relations (Chen et al., 2018b). Many recent efforts adopt a Siamese encoding architecture, where encoders based on convolutional neural networks (CNN) (Hu et al., 2014; Yin and Schütze, 2015) and recurrent neural networks (RNN) (Mueller and Thyagarajan, 2016) are widely used. A binary classifier is then stacked to the sequence pair encoder for the detection of a discourse relation.

In contrast to sentences, proteins are profiled in sequences with more intractable patterns, as well as in a drastically larger range of lengths. Precisely capturing the PPI requires much more comprehensive learning architectures to distill the latent information from the entire sequences, and to preserve the long-term ordering information. One recent work (Hashemifar et al., 2018), DPPI, uses a deep CNN-based architecture which focuses on capturing local features from protein profiles. DPPI represents the first work to deploy deep-learning to PPI prediction, and has achieved the state-of-the-art performance on the binary prediction task. However, it requires excessive efforts for data pre-processing such as constructing protein profiles by PSI-BLAST (Altschul et al., 1997), and does not incorporate a neural learning architecture that captures the important contextualized and sequential features. DNN-PPI (Li et al., 2018) represents another relevant work of this line, which deploys a different learning structure with two separated CNN encoders. However, DNN-PPI does not incorporate physicochemical properties into amino acid representations, and does not employ a Siamese learning architecture to fully characterize pairwise relations of sequences.

8.3 Methods

We introduce an end-to-end deep learning framework, PIPR, for sequence-based PPI prediction tasks. The overall learning architecture is illustrated in Fig 8.1. PIPR employs a Siamese architecture of residual RCNN encoder to better apprehend and utilize the mutual influence of two sequences. To capture the features of the protein sequences from scratch, PIPR pre-trains the embeddings of canonical amino acids to capture their contextual similarity and physicochemical properties. The latent representation of each protein in a protein pair is obtained by feeding the corresponding amino acid embeddings into the sequence encoder. The embeddings of these two sequences are then combined to form a sequence pair vector. Finally, this sequence pair vector is fed into a multi-layer perceptron with appropriate loss functions, suiting for specific prediction tasks. In this section, we describe the details of each model component. We begin with the denotations and problem specifications.

8.3.1 Preliminary

We use A to denote the vocabulary of 20 canonical amino acids. A protein is profiled as a sequence of amino acids $S = [a_1, a_2, ..., a_l]$ such that each $a_i \in A$. For each amino acid a_i, we use bold-faced

$\mathbf{a}_i$ to denote its embedding representation, which we are going to specify in Section 8.3.2.2. We use I to denote the set of protein pairs, and $p = (S_1, S_2) \in I$ denotes a pair of proteins of which our framework captures the interaction.

We address three challenging PPI prediction tasks based only on the primary sequence information: (i) *Binary prediction* seeks to provide a binary classifier to indicate whether the corresponding protein pair interacts, which is the simplest and widely considered problem setting in previous works (Hashemifar et al., 2018; Skrabanek et al., 2008; Sun et al., 2017a). (ii) *Interaction type prediction* is a multi-class classification problem, which seeks to identify the interaction type of two proteins. (iii) *Binding affinity estimation* aims at producing a regression model to estimate the strength of the binding interaction.

8.3.2 RCNN-based Protein Sequence Encoder

We employ a deep Siamese architecture of Residual RCNN to capture latent semantic features of the protein sequence pairs.

8.3.2.1 Protein Sequence Encoding

The RCNN seeks to leverage both the global sequential information and local features that are significant to the characterization of PPI from the protein sequences. This deep neural encoder stacks multiple instances of two computational modules, i.e. *convolution layers with pooling* (Section 2.3.1) and *bidirectional residual gated recurrent units* (Section 2.3.2). Fig. 8.2 shows an RCNN unit is shown on the left, and shows the entire structure of our RCNN encoder on the right. The RCNN encoder $E_{RCNN}(S)$ alternately stacks multiple occurrences of the above two intermediary neural network components. A convolution layer serves as the first encoding layer to extract local features from the input sequence. On top of that, a residual GRU layer takes in the preserved local features, whose outputs are passed to another convolution layer. Repeating of these two components in the network structure conducts an automatic multi-granular feature aggregation process on the protein sequence, while preserving the sequential and contextualized information on each granularity of the selected features. The last residual GRU layer is followed by another convolution layer for a final round of local feature selection to produce the last hid-

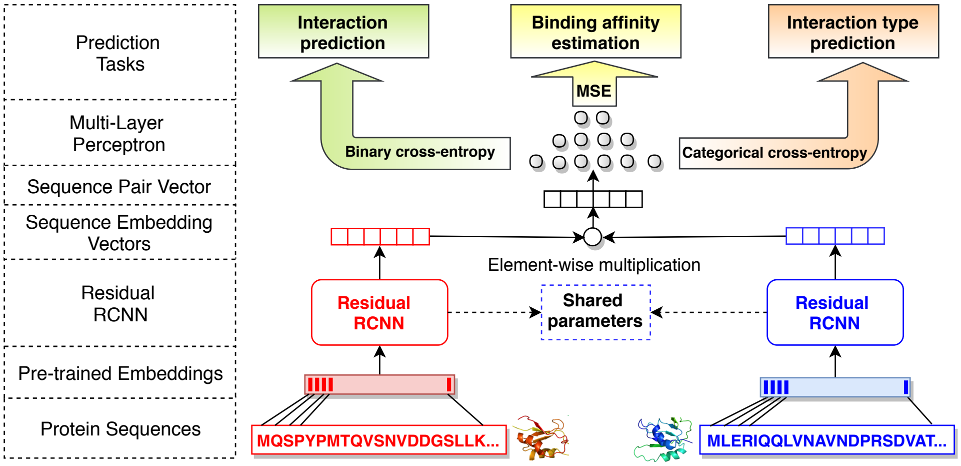


Figure 8.1: The overall learning architecture of our framework.

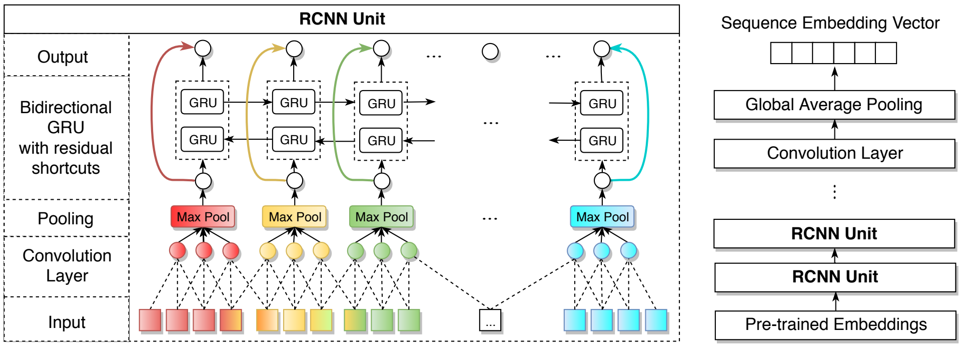


Figure 8.2: The structure of our residual RCNN encoder is shown on the right, and the RCNN unit is shown on the left. Each RCNN unit contains a convolution-pooling layer followed a bidirectional residual GRU.

den states $H' = [\mathbf{h}'_1, \mathbf{h}'_2, ..., \mathbf{h}'_{|H'|}]$. Note that the dimensionality of the last hidden states does not need to equal that of the previous hidden states. A high-level sequence embedding of the entire protein sequence is obtained from the global average-pooling (Lin et al., 2013) of H', i.e. $E_{RCNN}(S) = \frac{1}{|H'|} \sum_{i=1}^{|H'|} \mathbf{h}'_i$.

8.3.2.2 Pre-trained Amino Acid Embeddings

To support inputting the non-numerical sequence information, we provide a useful embedding method to represent each amino acid $a \in A$ as a semi-latent vector $\mathbf{a}$. Each embedding vector is a concatenation of two sub-embeddings, i.e. $\mathbf{a} = [\mathbf{a}_c, \mathbf{a}_{ph}]$.

The first part $\mathbf{a}_c$ measures the co-occurrence similarity of the amino acids, which is obtained by pre-training the Skip-Gram model (Mikolov et al., 2013b) on protein sequences. The learning objective of Skip-Gram is to minimize the following negative log likelihood loss.

$$J_{SG} = -\frac{1}{|S|} \sum_{a_t \in S} \sum_{-C<j<C} \log p(\mathbf{a}_{c,t+j}|\mathbf{a}_{c,t})$$

$\mathbf{a}_{c,t}$ thereof is the first-part embedding of the t-th amino acid $a_t \in S$, $\mathbf{a}_{c,t+j}$ is that of a neighboring amino acid, and C is the size of half context[1]. The probability p is defined as the following softmax:

$$p(\mathbf{a}_{c,t+j}|\mathbf{a}_{c,t}) = \frac{\exp(\mathbf{a}_{c,t+j} \cdot \mathbf{a}_{c,t})}{\sum_{k=1}^{n} \exp(\mathbf{a}'_{c,k} \cdot \mathbf{a}_{c,t})}$$

where n is the negative sampling size, and $\mathbf{a}'_{c,k}$ is a negative sample that does not co-occur with $\mathbf{a}_{c,t}$ in the same context.

The second part $\mathbf{a}_{ph}$ represents the similarity of electrostaticity and hydrophobicity among amino acids. The 20 amino acids can be clustered into seven classes based on their dipoles and volumes of the side chains to reflect this property. Thus, $\mathbf{a}_{ph}$ is a one-hot encoding based on the classification defined by Shen et al. (2007).

8.3.3 Learning Architecture and Learning Objectives

Our framework characterizes the interactions in the following two stages.

8.3.3.1 Siamese Architecture

Given a pair of proteins $p = (S_1, S_2) \in I$, the same RCNN encoder is used to obtain the sequence embeddings $E_{RCNN}(S_1)$ and $E_{RCNN}(S_2)$ of both proteins. Both sequence embeddings are combined using element-wise multiplication, i.e., $E_{RCNN}(S_1) \odot E_{RCNN}(S_2)$. This is a commonly used operation to infer the relation of sequence embeddings (Hashemifar et al., 2018; Jiang et al., 2018a; Rocktäschel et al., 2016b; Tai et al., 2015). Note that some works use the concatenation of sequence embeddings (Sun et al., 2017a; Yin and Schütze, 2015) instead of multiplication, which we find to be less effective in modeling the symmetric relations of proteins.

[1] The context of Skip-Gram means a subsequence of a given protein sequence S, such that the subsequence is of $2C + 1$ length.

8.3.3.2 Learning Objectives

A multi-layer perceptron (MLP) with leaky ReLU (Maas et al., 2013) is applied to the previous sequence pair representation, whose output $\hat{s}^p$ is either a vector or a scalar, depending on whether the model solves a classification or a regression task for the protein pair p. The entire learning architecture is trained to optimize the following two types of losses according to different PPI prediction problems.

(i) *Cross-entropy loss* is optimized for the two classification problems, i.e. binary prediction and interaction type prediction. In this case, the MLP output $\hat{s}^p$ is a vector, whose dimensionality equals the number of classes m. $\hat{s}^p$ is normalized by a softmax function, where the i-th dimension $s_i^p = \frac{\exp(\hat{s}_i^p)}{\sum_j \exp(\hat{s}_j^p)}$ corresponds to the confidence score for the i-th class. The learning objective is to minimize the following cross-entropy loss, where c^p is a one-hot indicator for the class label of protein pair p.

$$L^{(1)} = -\frac{1}{|I|} \sum_{p \in I} \sum_{i=1}^{m} c_i^p \log s_i^p$$

(ii) *Mean squared loss* is optimized for the binding affinity estimation task. In this case, $\hat{s}^p$ is a scalar output that is normalized by a sigmoid function $s^p = \frac{1}{1+\exp(\hat{s}^p)}$, which is trained to approach the normalized ground truth score $c^p \in [0, 1]$ by minimizing the following objective function:

$$L^{(2)} = \frac{1}{|I|} \sum_{p \in I} |s^p - c^p|^2$$

8.4 Experiments

We present the experimental evaluation of the proposed framework on three PPI prediction tasks, i.e. binary prediction, multi-class interaction type prediction, and binding affinity estimation. The experiments are conducted on the following datasets.

8.4.1 Datasets

Guo's Datasets. Guo et al. (2008) generate several datasets from different species for the binary prediction of PPIs. Each dataset contains a balanced number of positive and negative samples. Among these resources, the *Yeast dataset* is a widely used benchmark by most state-of-the-art methods (Hashemifar et al., 2018; Wong et al., 2015; You et al., 2013, 2014). There are 2,497 proteins forming 11,188 cases of PPIs, with half of them representing the positive cases, and the other half the negative cases. The positive cases are selected from the database of interacting proteins DIP_20070219 (Salwinski et al., 2004), where proteins with fewer than 50 amino acids or $\geq 40\%$ sequence identity are excluded. We use the full protein sequences in our model, which are obtained from the UniProt (Consortium et al., 2018). The negative cases are generated by randomly pairing the proteins without evidence of interaction, and filtered by their sub-cellular locations. In other words, non-interactive pairs residing in the same location are excluded.

In addition, we combine the data for *C.elegans*, *E.coli*, and *Drosophila* as the *multi-species dataset*. We use the cluster analysis of the CD-HIT (Li and Godzik, 2006) program to generate non-redundant subsets. Proteins with fewer than 50 amino acids or high sequence identify (40%, 25%, 10%, or 1%) are removed.

STRING Datasets. The STRING database (Szklarczyk et al., 2016) annotates PPIs with their types. There are seven types of interactions: activation, binding, catalysis, expression, inhibition, post-translational modification (ptmod), and reaction. We download all interaction pairs for *Homo sapiens* from database version 10.5 (Szklarczyk et al., 2016), along with their full protein sequences. Among the corresponding proteins, we randomly select 3,000 proteins and 8,000 proteins that share less than 40% of sequence identity to generate two subsets. In this process, we randomly sample instances of different interaction types to ensure a balanced class distribution. Eventually, the two generated datasets, denoted by SHS27k and SHS148k, contain 26,945 cases and 148,051 cases of interactions respectively. We use these two datasets for the PPI type prediction task.

SKEMPI Dataset. We obtain the protein binding affinity data from SKEMPI (the Structural database of Kinetics and Energetics of Mutant Protein Interactions) (Moal and Fernández-Recio,

2012) for the affinity estimation task. It contains 3,047 binding affinity changes upon mutation of protein sub-units within a protein complex. The binding affinity is measured by equilibrium dissociation constant (K_d), reflecting the strength of biomolecular interactions. The smaller K_d value means the higher binding affinity. Each protein complex contains single or multiple amino acid substitutions.The sequence of the protein complex is retrieved from the Protein Data Bank (PDB) (Berman et al., 2000). We manually replace the mutated amino acids. For duplicate entries, we take the average K_d. The final dataset results in the binding affinity of 2,792 mutant protein complexes, along with 158 wild-types.

8.4.2 Binary PPI Prediction

Binary PPI prediction is the primary task targeted by a handful of previous works (Hashemifar et al., 2018; Shen et al., 2007; Sun et al., 2017a; Yang et al., 2010; You et al., 2015). The objective of these works is to identify whether a given pair of proteins interacts or not based on their sequences. We evaluate PIPR based on Guo's datasets. The Yeast benchmark dataset thereof is used to compare PIPR with various baseline approaches, and the multi-species dataset is to demonstrate PIPR's capability of predicting interactions for proteins of different species that share very low sequence identity with those in training.

The baseline approaches include SVM-AC (Guo et al., 2008), kNN-CTD (Yang et al., 2010), EELM-PCA (You et al., 2013), SVM-MCD (You et al., 2014), MLP (Du et al., 2017), Random Forest LPQ (RF-LPQ) (Wong et al., 2015), SAE (Sun et al., 2017a), DNN-PPI (Li et al., 2018) and DPPI (Hashemifar et al., 2018). In addition, we report the results of a Siamese Residual GRU (SRGRU) architecture, which is a simplification of PIPR, where we discard all intermediary convolution layers and keep only the bidirectional residual GRU. The purpose of SRGRU is to show the significance of the contextualized and sequential information of protein profiles in characterizing PPIs. We also report the results of Siamese CNN (SCNN) by removing the residual GRU in PIPR. This degenerates our framework to a similar architecture to DPPI, but differs in that SCNN directly conducts an end-to-end training on raw sequences instead of requiring the protein profiles constructed by PSI-BLAST.

Table 8.1: Evaluation of binary PPI prediction on the Yeast dataset based on 5-fold cross-validation. We report the mean and standard deviation for the test sets.

Methods	**Accuracy (%)**	**Precision (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1-score (%)**	**MCC(%)**
SVM-AC	87.35 ± 1.38	87.82 ± 4.84	87.30 ± 5.23	87.41 ± 6.33	87.34 ± 1.33	75.09 ± 2.51
kNN-CTD	86.15 ± 1.17	90.24 ± 1.34	81.03 ± 1.74	NA	85.39 ± 1.51	NA
EELM-PCA	86.99 ± 0.29	87.59 ± 0.32	86.15 ± 0.43	NA	86.86 ± 0.37	77.36 ± 0.44
SVM-MCD	91.36 ± 0.4	91.94 ± 0.69	90.67 ± 0.77	NA	91.3 ± 0.73	84.21 ± 0.66
MLP	94.43 ± 0.3	96.65 ± 0.59	92.06 ± 0.36	NA	94.3 ± 0.45	88.97 ± 0.62
RF-LPQ	93.92 ± 0.36	96.45 ± 0.45	91.10 ± 0.31	NA	93.7 ± 0.37	88.56 ± 0.63
SAE	67.17 ± 0.62	66.90 ± 1.42	68.06 ± 2.50	66.30 ± 2.27	67.44 ± 1.08	34.39 ± 1.25
DNN-PPI	76.61 ± 0.51	75.1 ± 0.66	79.63 ± 1.34	73.59 ± 1.28	77.29 ± 0.66	53.32 ± 1.05
DPPI	94.55	96.68	92.24	NA	94.41	NA
SRGRU	93.77 ± 0.84	94.60 ± 0.64	92.85 ± 1.58	94.69 ± 0.81	93.71 ± 0.85	87.56 ± 1.67
SCNN	95.03 ± 0.47	95.51 ± 0.77	94.51 ± 1.27	95.55 ± 0.77	95.00 ± 0.50	90.08 ± 0.93
PIPR	**97.09 ± 0.24**	**97.00 ± 0.65**	**97.17 ± 0.44**	97.00 ± 0.67	**97.09 ± 0.23**	**94.17 ± 0.48**

Table 8.2: Statistical assessment (*t*-test; two-tailed) on the accuracy of binary PPI prediction. The statistically significant differences are highlighted in red.

p-value	**SRGRU**	**SCNN**	**PIPR**
SVM-AC	9.69E-05	1.22E-04	9.69E-05
kNN-CTD	1.03E-05	2.23E-05	2.84E-05
EELM-PCA	2.33E-05	3.94E-08	2.43E-10
SVM-MCD	1.67E-03	2.60E-06	1.35E-07
MLP	1.71E-01	5.29E-02	1.12E-06
RF-LPQ	7.28E-01	4.10E-03	1.75E-06
SAE	4.27E-10	1.78E-10	4.19E-09
DNN-PPI	1.62E-08	2.27E-10	2.70E-09
SRGRU	NA	2.87E-02	6.60E-04
SCNN	2.87E-02	NA	1.80E-04

We use AMSGrad (Reddi et al., 2018) to optimize the cross-entropy loss, for which we set the learning rate α to 0.001, the exponential decay rates β_1 and β_2 to 0.9 and 0.999, and batch size to 256 on both datasets. The number of occurrences for the RCNN units (i.e., one convolution-pooling layer followed by one bidirectional residual GRU layer) is set to 5, where we adopt 3-max-pooling and the convolution kernel of size 3. We set the hidden state size to be 50, and the RCNN output size to be 100. We set this configuration to ensure the RCNN to compress the selected features in a reasonably small vector sequence, before the features are aggregated by the last global average-pooling. We zero-pad short sequences to the longest sequence length in the dataset. This is a widely adopted technique for sequence modeling in NLP (Chen et al., 2018b;

He et al., 2015; Hu et al., 2014; Yin et al., 2016a; Zhou et al., 2017) as well as in bioinformatics (Min et al., 2017; Müller et al., 2018; Pan and Shen, 2018) for efficient training. Note that the configuration of embedding pre-training is discussed in Section 8.4.5, and the model configuration study of different hyperparameter values is provided in Section . All model variants are trained until converge at each fold of the cross-validation.

Evaluation protocol. Following the settings in previous works (Hashemifar et al., 2018; Shen et al., 2007; Sun et al., 2017a; You et al., 2015, 2014), we conduct 5-fold cross-validation (CV) on the Yeast dataset. Under the k-fold CV setting, the data is equally divided into k non-overlapping subsets, and each subset has a chance to train and to test the model so as to ensure an unbiased evaluation. We aggregate fix metrics on the test cases of each fold, i.e. the overall *accuracy*, *precision*, *sensitivity*, *specificity*, *F1*, and *Matthews correlation coefficient (MCC)* on positive cases. All these metrics are preferred to be higher to indicate better performance. Based on the reported accuracy over 5-folds, we also conduct two-tailed Welch's t-tests (Welch, 1947) to evaluate the significance of the improvement on different pairs of approaches. The *p*-values are adjusted by the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) to control the false discovery rate for multiple hypothesis testing.

Results. As shown in Table 8.1, the CNN-based architecture, DPPI, demonstrates state-of-the-art performance over other baselines that employ statistical learning algorithms or densely connected MLP. This shows the superiority of deep-learning-based techniques[2] in encapsulating various types of information of a protein pair, such as amino acid composition and their co-occurrences, and automatically extracting the robust ones for the learning objectives. That said, DPPI requires an extensive effort in data pre-processing, specifically in constructing the protein profile for each sequence. On average, each PSI-BLAST search of a protein against the NCBI non-redundant protein database (184,243,125 sequences) requires around 90 minutes of computation on our server. Even with eight cores, each search finishes in 15 minutes. We estimate that processing 2,497 sequences

[2] We are unable to obtain the source codes of two deep-learning methods, SAE and DNN-PPI. We implement these two models following the descriptions in their papers. Our implementations are verified by achieving comparable performance on the Pan's dataset (Pan et al., 2010) as reported in the papers. However, these two implementations can only achieve 67.17% and 76.61% in overall accuracy respectively on the Yeast dataset.

Table 8.3: Evaluation of binary PPI prediction on variants of multi-species (C. elegan, Drosophila, and E. coli) dataset.

Seq. Identity	**# of Proteins**	**Pos. Pairs**	**Neg. Pairs**	**Accuracy (%)**	**F1-Score (%)**
Any	11529	32959	32959	98.19	98.17
<0.40	9739	25916	22012	98.29	98.28
<0.25	7790	19458	15827	97.91	98.08
<0.10	5769	12641	9819	97.54	97.79
<0.01	5171	10747	8065	97.51	97.80

of the Yeast dataset from scratch can take about 26 days. It is worth mentioning that PIPR only requires 8 seconds to pre-train the amino acid embedding, and 2.5 minutes to train on the Yeast dataset (see Table 8.7). We implement SCNN to evaluate the performance of a simplified CNN architecture, which produces comparable results as DPPI. These two frameworks show that CNN can already leverage the significant features from primary protein sequences.

In addition, the SRGRU architecture has offered comparable performance to SCNN. This indicates that preserving the sequential and contextualized features of the protein sequences is as crucial as incorporating the local features. By integrating both significant local features and sequential information, PIPR outperforms DPPI by 2.54% in accuracy, 4.93% in sensitivity, and 2.68% in F1-Score. Next, we evaluate whether the improved accuracy of PIPR is statistically significant. Table 8.2 reports the p-values of SRGRU, SCNN, and PIPR compared to other baseline approaches, where the statistically significant comparisons (p-values < 0.01) are highlighted in red. Since the standard deviation of DPPI is unavailable, we are not able to include DPPI in this analysis. The evaluation shows that PIPR performs statistically significantly better than all other approaches, including SCNN and SRGRU. On the other hand, SCNN is not statistically significantly better than SRGRU. Thus, the residual RCNN is very promising for modeling binary PPIs.

We also report the 5-fold CV performance of PIPR on variants of the multi-species dataset, where proteins are excluded based on different thresholds of sequence identity. The results in Table 8.3 show that PIPR performs consistently well under lenient and stringent criteria of sequence identity between training and testing. More importantly, PIPR is able to train and test on multiple species, and is robust against extremely low sequence identity of less than 1%.

Table 8.4: Accuracy (%) and fold changes over zero rule for PPI interaction type prediction on two STRING datasets based on 10-fold cross-validation.

Features	N/A		AC					CTD					Embedded raw seqs		
Methods	Rand	Zero rule	SVM	RF	AdaBoost	kNN	Logistic	SVM	RF	AdaBoost	kNN	Logistic	SCNN	SRGRU	PIPR
SHS27k	14.28	16.70	33.17	44.82	28.67	35.44	25.47	35.56	45.76	31.81	35.56	30.57	55.54	51.06	**59.56**
(fold×)	—	1.00×	1.99×	2.68×	1.72×	2.12×	1.52×	2.13×	2.74×	1.90×	2.13×	1.83×	3.33×	3.06×	**3.57**×
SHS148k	14.28	16.21	28.17	36.01	27.87	33.81	24.96	31.37	36.65	29.67	33.13	26.96	55.29	54.05	**61.91**
(fold×)	—	1.00×	1.74×	2.22×	1.72×	2.09×	1.54×	1.94×	2.26×	1.83×	2.04×	1.66×	3.41×	3.33×	**3.82**×

Table 8.5: Results for binding affinity prediction on the SKEMPI dataset. Each measurement is an average of the test sets over 10-fold cross-validation.

Features	AC				CTD				Embedded raw seqs		
Methods	BR	SVM	RF	AdaBoost	BR	SVM	RF	AdaBoost	SCNN	SRGRU	PIPR
MSE($\times 10^{-2}$)	1.70	2.20	1.77	1.98	1.86	1.84	1.49	1.84	0.87	0.95	**0.63**
MAE($\times 10^{-2}$)	9.56	11.81	9.81	11.15	10.20	11.04	9.06	10.69	6.49	7.08	**5.48**
Corr	0.564	0.353	0.546	0.451	0.501	0.501	0.640	0.508	0.831	0.812	**0.873**

8.4.3 Interaction Type Prediction

The objective of this task is to predict the interaction type of two interacting proteins. We evaluate this task based on SHS27k and SHS148k datasets. To the best of our knowledge, much fewer efforts attempt for the multi-class PPI prediction in contrast to the binary prediction. Zhu et al. (2006) train a two-stage SVM classifier to distinguish obligate, non-obligate, and crystal packing interactions; Silberberg et al. (2014) use logistic regression to predict several types of enzymatic actions. However, none of their implementations are publicly available. Different from the categories of interaction types used above, we aim at predicting the interaction types annotated by the STRING database.

We train several statistical learning algorithms on the widely employed AC and CTD features for protein characterization as our baselines. These algorithms include SVM, Random Forest, Adaboost (SAMME.R algorithm (Zhu et al., 2009)), kNN classifier, and logistic regression. For deep-learning-based approaches, we deploy the SCNN architecture where an output MLP with categorical cross-entropy loss is incorporated, as well as a similar SRGRU architecture into comparison. Results of two naïve baselines of random guessing and zero rule (i.e., simply predicting the majority class) are also reported for reference.

Evaluation protocol. All approaches are evaluated on the two datasets by 10-fold CV, using the same partition scheme for a more unbiased evaluation (James et al., 2013; McLachlan et al., 2005). We carry forward the model configurations from the last experiment to evaluate the performance of the frameworks under controlled variables. For baseline models, we examine three different ways of combining the feature vectors of the two input proteins, i.e. element-wise multiplication, the Manhattan difference (i.e. the absolute differences of corresponding features (Mueller and Thyagarajan, 2016)) and concatenation. The Manhattan difference consistently obtains better performance, considering the small values of the input features and the asymmetry of the captured protein relations.

Results. The prediction accuracy and fold changes over the zero rule baseline are reported in Table 8.4. Note that since the multi-class prediction task is much more challenging than the binary prediction task, it is expected to observe lower accuracy and longer training-time (Table 8.7) than that reported in the previous experiment. Among all the baselines using explicit features, the CTD-based models perform better than the AC-based ones. CTD descriptors seek to cover both continuous and discontinuous interaction information (Yang et al., 2010), which potentially better discriminate among PPI types.

The best baseline using Random Forest thereof achieves satisfactory results by more than doubling the accuracy of zero rule on the smaller SHS27k dataset. However, on the larger SHS148k dataset, the accuracy of these explicit-feature-based models is notably impaired. We hypothesize that such predefined explicit features are not representative enough to distinguish the PPI types. On the other hand, the deep-learning-based approaches do not need to explicitly utilize these features, and perform consistently well in both settings. The raw sequence information is sufficient for these approaches to drastically outperform the Random Forest by at least 5.30% in accuracy on SHS27k and 17.40% in accuracy on SHS148k. SCNN thereof outperforms SRGRU by 4.48% and 1.24% in accuracy on SHS27k and SHS148k, respectively. This implies that the local interacting features are relatively more deterministic than contextualized and sequential features on this task. The results by the residual RCNN-based framework are very promising, as it outperforms SCNN by 4.02% and 6.62% in accuracy on SHS27k and SHS148k respectively. It also remarkably outperforms the best explicit-feature-based baselines on the two datasets by 13.80% and 25.26% in accuracy, and more than 3.5 of fold changes over the zero rule on both datasets.

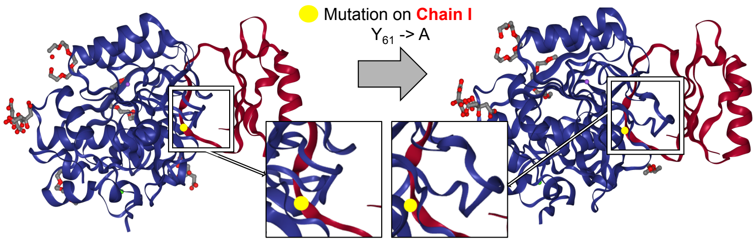
8.4.4 Binding Affinity Estimation

Lastly, we evaluate PIPR for binding affinity estimation using the SKEMPI dataset. We employ the mean squared loss variant of PIPR to address this regression task. Since the lengths of protein sequences in SKEMPI are much shorter than those in the other datasets, we accordingly reduce the occurrences of RCNN units to 3, while other configurations remain unchanged. For baselines, we compare against several regression models based on the AC and CTD features, which include Bayesian Redge regressor (BR), SVM, Adaboost with decision tree regressors and Random Forest regressor. The corresponding features for two sequences are again combined via the Manhattan difference. We also modify SCNN and SRGRU to their mean squared loss variants, in which we reduce the layers in the same way of RCNN.

Evaluation protocol. We aggregate three metrics through 10-fold CV, i.e. *mean squared error* (*MSE*), *mean absolute error* (*MAE*) and *Pearson's correlation coefficient* (*Corr*). These are three commonly reported metrics for regression tasks, for which lower *MSE* and *MAE* as well as higher *Corr* indicate better performance. In the cross-validation process, we normalize the affinity values of the SKEMPI dataset to $[0, 1]$ via min-max re-scaling[3].

Results. Table 8.5 reports the results for this experiment. It is noteworthy that, one single change of amino acid can lead to a drastic effect on binding affinity. While such subtle changes are difficult to be reflected by the explicit features, the deep-learning-based methods can competently capture such changes from the raw sequences. Our RCNN-based framework again offers the best performance among the deep-learning-based approaches, and significantly outperforms the best baseline (CTD-based Random Forest) by offering a 0.233 increase in *Corr*, as well as remarkably lower *MSE* and *MAE*. Figure 8.3 demonstrates an example of the effect of changing an amino acid in a protein complex. Tyrosine at position 61 of Chymotrypsin inhibitor 2 (Chain I) is substituted with Alanine, causing the neighboring region of Subtilisin BPN' precursor (Chain E) to relax.

[3] This is due to that we use sigmoid function to smooth the output of the regressor. Note that this process does not affect correlation, while MSE, MAE and the original affinity scores can be easily re-scaled back.



PDB ID	1TM1 (wild type)	1TO1 (mutant)
Binding Affinity (k_d)	2.24E-12	2.70E-10
PIPR Prediction (k_d)	2.54E-12	3.52E-10

Figure 8.3: Mutation effects on structure and binding affinity. The blue entity is Subtilisin BPN' precursor (Chain E), and the red entity is Chymotrypsin inhibitor (Chain I). The mutation is highlighted in yellow. The wild type (1TM1) and mutant (1TO1) complexes are retrieved from PDB.

The binding affinity (k_d) changes from 2.24E-12 to 2.70E-10, which is validly captured by PIPR. While our experiment is conducted on a relatively small dataset, we seek to extend our PIPR framework to a more generalized solution for binding affinity estimation, once a larger and more heterogeneous corpus is available.

8.4.5 Amino Acid Embeddings

We further investigate the settings of amino acid embeddings in this subsection. Each amino acid is represented by a vector of numerical values that describe its relative physicochemical properties. The first part of the embedding vector $\mathbf{a}_c$, which measures the co-occurrence similarity of the amino acids in protein sequences, is empirically set as a 5-dimensional vector. $\mathbf{a}_c$ is obtained by pre-training the Skip-Gram model on all 8,000 sequences from our largest STRING dataset, SHS148k, using a context window size of 7 and a negative sampling size of 5. The second part contains a 7-dimensional vector, $\mathbf{a}_{ph}$, which describes the categorization of electrostaticity and hydrophobicity for the amino acid. We examine the performance of using each part individually, as well as the performance of combining them as used in our framework. In addition, we include a naïve one-hot vector representation, which does not consider the relatedness of amino acids and treats each of them independently. Table 8.6 shows that, once we remove either of the two parts of the proposed embedding, the performance of the model slightly drops. Meanwhile, the proposed pre-trained embeddings lead to noticeably better performance of the model than adopting the naïve one-hot encodings of the canonical amino acids. This pre-training process completes in 8 seconds

Table 8.6: Comparison of amino acid representations based on binary prediction.

	$[\mathbf{a}_c, \mathbf{a}_{ph}]$	$\mathbf{a}_c$ only	$\mathbf{a}_{ph}$ only	One-hot
Dimension	12	5	7	20
Accuracy	**97.09**	96.67	96.03	96.11
Precision	**97.00**	96.35	95.91	96.34
F1-Score	**97.09**	96.51	96.08	96.10

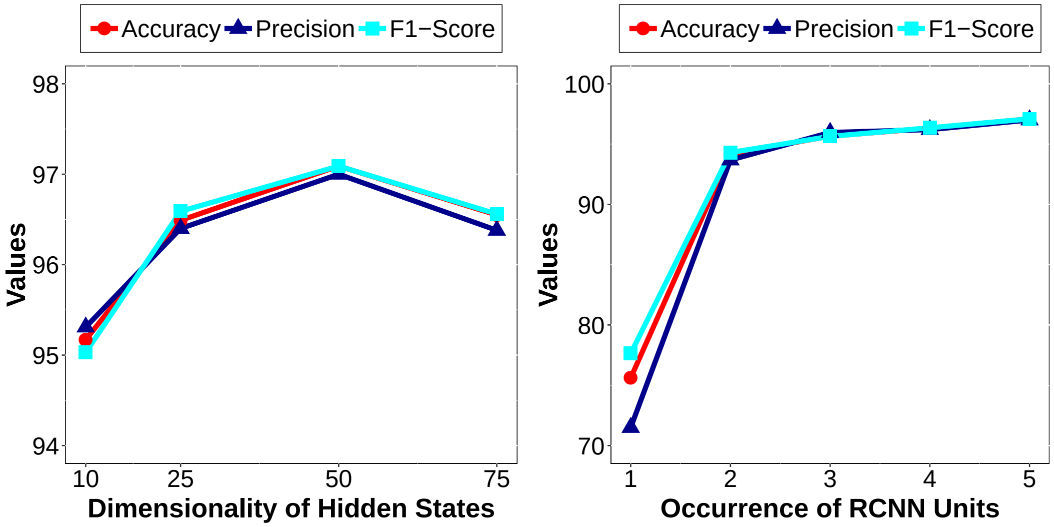
Table 8.7: Run-Time of training embeddings and different prediction tasks.

Task	**Embeddings**	**Binary**	**Multi-class**	**Multi-class**	**Regression**
Dataset	**SHS148k**	**Yeast**	**SHS27k**	**SHS148k**	**SKEMPI**
Sample Size	8,000	11,188	26,945	148,051	2,950
Training Time	8sec	2.5min	15.8min	138.3min	12.5min

on a commodity workstation as shown in Table 8.7. This is a one-time effort that can be reused on different tasks and datasets.

8.4.6 Hyperparameter Study

We examine the configuration of two critical factors that can affect the performance of our framework: the dimensionality of hidden states and the number of occurrences for the RCNN units. We show the effects of different settings of these two factors based on the binary PPI prediction task. The hidden state sizes are chosen from $\{10, 25, 50, 75\}$. As illustrated in Fig 8.4a, the performance of PIPR initially increases as we raise the dimensionality of the hidden states until it passes 50, and then starts to decline. The occurrences of RCNN units contribute to the levels of granularity in feature aggregation. Fewer occurrences correspond to less aggregation. However, too many occurrences can lead to over-compressing the features. We examine the occurrences from 1 to 5 based on Yeast. Note that we do not adopt the setting with 6 occurrences, where the RCNN encoder over-compresses the extracted features to a very small number of latent vectors before the last global average pooling. Aligned with our hypothesis, Fig 8.4b shows that the accuracy, precision, and F1-score improve when we increase the number of occurrences of the RCNN units. The improvement from 2 to 5 occurrences is marginal, which shows that our framework is robust to this setting as long as there are more than 2 occurrences of RCNN units.



(a) Hidden States

(b) RCNN Units

Figure 8.4: Performance evaluation on dimensionality of hidden states, and the number of occurrences of the RCNN units.

8.4.7 Run-time Analysis

All of the experiments are conducted on one NVIDIA GeForce GTX 1080 Ti GPU. We report the training time for each experiment, as well as for the amino acid embedding in Table 8.7. For each experiment, we calculate the average training time over either 5-fold (Yeast dataset) or 10-fold (others) CV. In both binary and multi-class predictions, the training time increases along with the increased number of training cases. The regression estimation generally requires more iterations per training case to converge than classification tasks. Thus, with much fewer cases, the training time on SKEMPI for affinity estimation is more than that on the Yeast dataset for binary prediction.

8.5 Conclusion

In this chapter, we propose a novel end-to-end framework for PPI prediction based on the amino acid sequences. Our proposed framework, PIPR, employs a residual RCNN, which provides an automatic multi-granular feature selection mechanism to capture both local significant features and sequential features from the primary protein sequences. By incorporating the RCNN in a Siamese-based learning architecture, the framework captures effectively the mutual influence of protein pairs, and generalizes well to address different PPI prediction tasks without the need for predefined features. Extensive experimental evaluations on five datasets show promising performance of our framework on three challenging PPI prediction tasks. This also leads to significant amelioration over various baselines. Experiments on datasets of different sizes also demonstrate satisfactory scalability of the framework. For future work, one important direction is to apply the PIPR framework to other sequence-based inference tasks in bioinformatics, such as modeling RNA and protein interactions. We also seek to incorporate attention mechanisms (Vaswani et al., 2017) to help pinpoint interaction sites on protein sequences, and apply PIPR to predict confidence of interactions in the form of ordinal regression. Since PIPR has alleviated any costly domain-invariant feature engineering process, how to extend PIPR with transfer learning based domain adaptation for different species is another meaningful direction.

CHAPTER 9

Learning Multi-granular Associations of Lexemes and Sentences

In this chapter, we present a novel approach for exploting the cross-lingual correspondence of lexemes and sentences, which seeks to enrich the knowledge of lexicographic knowledge bases.

9.1 Introduction

Cross-lingual semantic representation learning has attracted significant attention recently. Various approaches have been proposed to align words of different languages in a shared embedding space (Ruder et al., 2017). By offering task-invariant semantic transfers, these approaches critically support many cross-lingual NLP tasks including neural machine translations (NMT) (Devlin et al., 2014), bilingual document classification (Zhou et al., 2016), knowledge alignment (Chen et al., 2018c) and entity linking (Upadhyay et al., 2018).

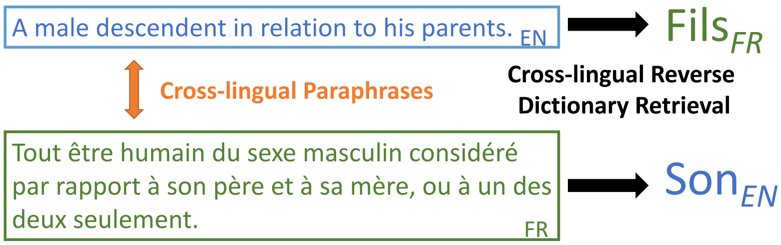
While many existing approaches have bee proposed to associate lexical semantics between languages (Chandar et al., 2014; Gouws et al., 2015; Luong et al., 2015a), modeling the correspondence between lexical and sentential semantics across different languages is still an unresolved challenge. We argue that learning to represent such cross-lingual and multi-granular correspondence is well desired and natural for multiple reasons. One reason is that, learning word-to-word correspondence has a natural limitation, considering that many words do not have direct translations in another language. For example, *schadenfreude* in German, which means *a feeling of joy* *that comes from knowing the troubles of other people*, has no proper English counterpart word. To appropriately learn the representations of such words in bilingual embeddings, we need to capture their meanings based on the definitions.

Besides, modeling such correspondence is also highly beneficial to many application scenarios. One example is cross-lingual semantic search of concepts (Hill et al., 2016), where the lexemes or concepts are retrieved based on sentential descriptions (see Fig. 9.1). Others include discourse relation detection in bilingual dialogue utterances (Jiang et al., 2018b), multilingual text summarization (Nenkova et al., 2012), and educational applications for foreign language learners. Finally, it is natural in foreign language learning that a human learns foreign words by looking up their meanings in the native language (Hulstijn et al., 1996). Therefore, learning such correspondence essentially mimics human learning behaviors.

However, realizing such a representation learning model is a non-trivial task, inasmuch as it requires a comprehensive learning process to effectively compose the semantics of arbitrary-length sentences in one language, and associate that with single words in another language. Consequently, this objective also demands high-quality cross-lingual alignment that bridges between single and sequences of words. Such alignment information is generally not available in the parallel and seed-lexicon that are utilized by bilingual word embeddings (Ruder et al., 2017).

To incorporate the representations of bilingual lexical and sentential semantics, we propose an approach to capture *the mapping from the definitions to the corresponding foreign words* by leveraging *bilingual dictionaries*[1]. The proposed model `BilDRL` (**Bil**ingual **D**ictionary **R**epresentation **L**earning) first constructs a word embedding space with pre-trained bilingual word embeddings. Based on cross-lingual word definitions, a sentence encoder is trained to realize the mapping from literal descriptions to target words in the bilingual word embedding space, for which we investigate with multiple encoding techniques. To enhance cross-lingual learning on limited resources, `BilDRL` conducts multi-task learning on different directions of a language pair. Moreover, `BilDRL` enforces a joint learning strategy of bilingual word embeddings and the sentence

[1] We refer the term *dictionary* to its regular meaning, i.e. lexical definitions of words. Note that this is different from some papers on bilingual settings that refer dictionaries to seed lexicons for one-to-one word mappings.



A male descendent in relation to his parents. EN
Fils$_{FR}$
Cross-lingual Paraphrases
Cross-lingual Reverse Dictionary Retrieval
Tout être humain du sexe masculin considéré par rapport à son père et à sa mère, ou à un des deux seulement. FR
Son$_{EN}$

Figure 9.1: An example illustrating the two cross-lingual tasks. The *cross-lingual reverse dictionary retrieval* finds cross-lingual target words based on descriptions. In terms of *cross-lingual paraphrases*, the French sentence (which means *any male being considered in relation to his father and mother, or only one of them*) describes the same meaning as the English sentence, but has much more content details.

encoder, which seeks to gradually adjust the embedding space to better suit the representation of cross-lingual word definitions.

To show the applicability of `BilDRL`, we conduct experiments on two useful cross-lingual tasks (see Fig. 9.1). (i) *Cross-lingual reverse dictionary retrieval* seeks to retrieve words or concepts given descriptions in another language. This task is useful to help users find foreign words based on the notions or descriptions, and is especially beneficial to users such as translators, foreigner language learners and technical writers using non-native languages. We show that `BilDRL` achieves promising results on this task, while bilingual multi-task learning and joint learning dramatically enhance the performance. (ii) *Bilingual paraphrase identification* asks whether two sentences in different languages essentially express the same meaning, which is critical to question answering or dialogue systems that apprehend multilingual utterances (Bannard and Callison-Burch, 2005). This task is challenging, as it requires a model to comprehend cross-lingual paraphrases that are inconsistent in grammar, content details and word orders. `BilDRL` maps sentences to the lexicon embedding space. This process reduces the problem to evaluate the similarity of lexicon embeddings, which can be easily solved by a simple classifier. `BilDRL` performs well with even a small amount of data, and significantly outperforms previous approaches.

9.2 Related Work

We discuss two lines of relevant work.

Bilingual word embeddings. Various approaches have been proposed for training bilingual word embeddings. These approaches span in two families: off-line mappings and joint training.

The off-line mapping based approach fixes the structures of pre-trained monolingual embeddings, and induces bilingual projections based on seed lexicons (Mikolov et al., 2013a). Some variants of this approach improve the quality of projections by adding constraints such as orthogonality of transforms, normalization and mean centering of embeddings (Artetxe et al., 2016; Vulić et al., 2016; Xing et al., 2015). Others adopt canonical correlation analysis to map separate monolingual embeddings to a shared embedding space (Doval et al., 2018; Faruqui and Dyer, 2014).

Unlike off-line mappings, joint training models simultaneously update word embeddings and cross-lingual alignment. In doing so, such approaches generally capture more precise cross-lingual semantic transfer (Ruder et al., 2017; Upadhyay et al., 2018). While a few such models still maintain separated embedding spaces for each language (Artetxe et al., 2017), more of them maintain a unified space for both languages. The cross-lingual semantic transfer by these models is captured from parallel corpora with sentential or document-level alignment, using techniques such as bilingual bag-of-words distances (BilBOWA) (Gouws et al., 2015), Skip-Gram (Coulmance et al., 2015) and sparse tensor factorization (Vyas and Carpuat, 2016).

Neural sentence modeling. Neural sentence models seek to capture phrasal or sentential semantics from word sequences. They often adopt encoding techniques such as recurrent neural encoders (RNN) (Kiros et al., 2015), convolutional encoders (CNN) (Chen et al., 2018b), and attentive encoders (Rocktäschel et al., 2016a) to represent the composed semantics of a sentence as an embedding vector. Recent works have focused on apprehending pairwise correspondence of sentential semantics by adopting multiple neural sentence models in one learning architecture, including Siamese models for detecting discourse relations of sentences (Sha et al., 2016), and sequence-to-sequence models for tasks like style transfer (Shen et al., 2017), text summarization (Chopra et al., 2016) and translation (Wu et al., 2016).

On the other hand, fewer efforts have been put to characterizing the associations between sentential and lexical semantics. Hill et al. (Hill et al., 2016) and Xie et al. (Xie et al., 2016) learn off-line mappings between monolingual descriptions and lexicons to capture such associations.

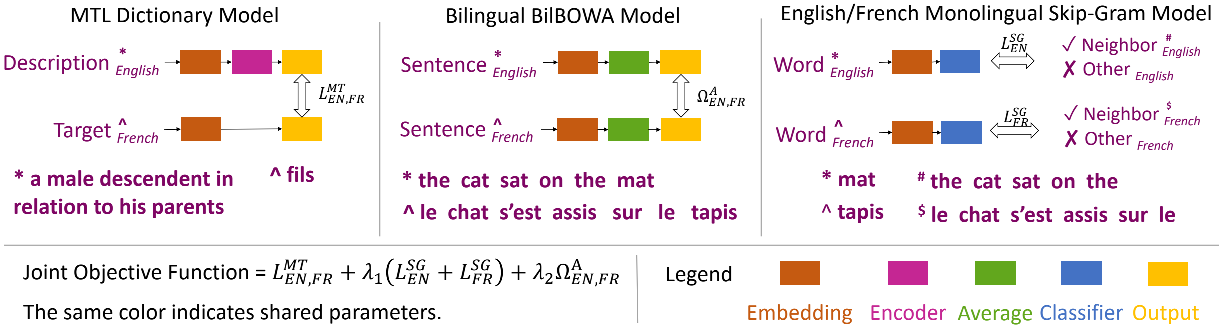


Figure 9.2: Joint learning architecture of `BilDRL`.

Eisner et al. (Eisner et al., 2016) adopt a similar approach to capture emojis based on descriptions.

At the best of our knowledge, there has been no previous approach to learn to discover the correspondence of sentential and lexical semantics in a multilingual scenario. This is exactly the focus of our work, in which the proposed strategies of multi-task learning and joint learning are critical to the corresponding learning process under limited resources. Utilizing such correspondence, our approach also sheds light on addressing discourse relation detection in a multilingual scenario.

9.3 Modeling Bilingual Dictionaries

We hereby begin our modeling with the formalization of bilingual dictionaries. We use $\mathcal{L}$ to denote the set of languages. For a language $l \in \mathcal{L}$, V_l denotes its vocabulary, where for each word $w \in V_l$, bold-faced $\mathbf{w} \in \mathbb{R}^k$ denotes its embedding vector. A l_i-l_j bilingual dictionary $D(l_i, l_j)$ (or simply D_{ij}) contains dictionary entries $(w^i, S^j_w) \in D_{ij}$, in which $w^i \in V_{l_i}$, and $S^j_w = w^j_1 \ldots w^j_n$ $(w^j_{\cdot} \in V_{l_j})$ is a cross-lingual definition that describes the word w^i with a sequence of words in language l_j. For example, a French-English dictionary $D(\mathrm{Fr}, \mathrm{En})$ could include a French word *appétite* accompanied by its English definition *desire for, or relish of food or drink*. Note that, for a word w^i, multiple definitions in l_j may coexist.

`BilDRL` is constructed and improved through three stages, as depicted in Fig. 9.2. A sentence encoder is first used to learn from a bilingual dictionary the association between words and definitions. Then in a pre-trained bilingual word embedding space, multi-task learning is conducted on both directions of a language pair. Lastly, joint learning with word embeddings is enforced to simultaneously adjust the embedding space during the training of the dictionary model, which further enhances the cross-lingual learning process.

It is noteworthy that, NMT (Wu et al., 2016) is considered as an ostensibly relevant method to ours. NMT does not apply to our problem setting due to that it has major differences from our work in those perspectives: (i) In terms of data modalities, NMT has to bridge between corpora of the same granularity, i.e. either between sentences or between lexemes. This is unlike `BilDRL` that captures multi-granular correspondence of semantics across different modalities, i.e. sentences and words; (ii) As for learning strategies, NMT relies on an encoder-decoder architecture using end-to-end training (Luong et al., 2015b), while `BilDRL` employs joint learning of a dictionary-based sentence encoder and a bilingual embedding space.

9.3.1 Encoders for Bilingual Dictionaries

`BilDRL` models a dictionary using a neural sentence encoder $E(S)$, which composes the meaning of the sentence into a latent vector representation. We hereby introduce this model component, which is designed to be a GRU encoder with self-attention as described in Section 2.3.3. We also experiment with other widely used neural sentence modeling techniques[2], which are however outperformed by the attentive GRU in our tasks. These techniques include the vanilla GRU, CNN (Kalchbrenner et al., 2014), and linear bag-of-words (BOW) (Hill et al., 2016).

9.3.2 Basic Learning Objective

The objective of learning the dictionary model is to map the encodings of cross-lingual word definitions to the target word embeddings. This is realized by minimizing the following L_2 loss,

$$L_{ij}^{\mathrm{ST}} = \frac{1}{|D_{ij}|} \sum_{(w^i, S_w^j) \in D_{ij}} \left\| E_{ij}(S_w^j) - \mathbf{w}^i \right\|_2^2$$

[2] Note that recent advances in monolingual contextualized embeddings like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) can also be supported to represent sentences for our setting. We leave them as future work, as they require non-trivial adaption to both multilingual settings and joint training, and extensive pre-training on external corpora.

in which E_{ij} is the dictionary model that maps from descriptions in l_j to words in l_i.

The above defines the basic model variants of `BilDRL` that learns on a single dictionary. For word representations in the learning process, `BilDRL` initializes the embedding space using pre-trained word embeddings. Note that, without adopting the joint learning strategy in Section 9.3.4, the learning process does not update word embeddings that are used to represent the definitions and target words. While other forms of loss such as cosine proximity (Hill et al., 2016) and hinge loss (Xie et al., 2016) may also be used in the learning process, we find that L_2 loss consistently leads to better performance in our experiments.

9.3.3 Bilingual Multi-task Learning

In cases where entries in a bilingual dictionary are not amply provided, learning the above bilingual dictionary on one ordered language pair may fall short in insufficiency of alignment information. One practical solution is to conduct a bilingual multi-task learning process. In detail, given a language pair (l_i, l_j), we learn the dictionary model E_{ij} on both dictionaries D_{ij} and D_{ji} with shared parameters. Correspondingly, we rewrite the previous learning objective function as below, in which $D = D_{ij} \cup D_{ji}$.

$$L_{ij}^{\mathrm{MT}} = \frac{1}{|D|} \sum_{(w,S_w)\in D} \|E_{ij}(S_w) - \mathbf{w}\|_2^2$$

This strategy non-trivially requests the same dictionary model to represent semantic transfer in two directions of the language pair. To fulfill such a request, we initialize the embedding space using the BilBOWA embeddings (Gouws et al., 2015), which provide a unified embedding space that resolves both monolingual and cross-lingual semantic relatedness of words. In practice, we find this simple multi-task strategy to bring significant improvement to our cross-lingual tasks. Note that, besides BilBOWA, other joint-training bilingual embeddings in a unified space (Doval et al., 2018) can also support this strategy, for which we leave the comparison to future work.

9.3.4 Joint Learning Objective

While above learning strategies are based on a fixed embedding space, we lastly propose a joint learning strategy. During the training process, this strategy simultaneously updates the embedding space based on both the dictionary model and the bilingual word embedding model. The learning is through asynchronous minimization of the following joint objective function,

$$J = L_{ij}^{\mathrm{MT}} + \lambda_1 (L_i^{\mathrm{SG}} + L_j^{\mathrm{SG}}) + \lambda_2 \Omega_{ij}^{\mathrm{A}}$$

where λ_1 and λ_2 are two positive hyperparameters. L_i^{SG} and L_j^{SG} are the original Skip-Gram losses (Mikolov et al., 2013c) to separately obtain word embeddings on monolingual corpora of l_i and l_j. Ω_{ij}^{A}, termed as below, is the alignment loss to minimize bag-of-words distances for aligned sentence pairs (S^i, S^j) in parallel corpora C_{ij}.

$$\Omega_{ij}^{\mathrm{A}} = \frac{1}{|C_{ij}|} \sum_{(S^i, S^j) \in C_{ij}} \left\| \frac{1}{|S^i|} \sum_{w_m^i \in S^i} \mathbf{w}_m^i - \frac{1}{|S^j|} \sum_{w_n^j \in S^j} \mathbf{w}_n^j \right\|_2^2$$

The joint learning process adapts the embedding space to better suit the dictionary model, which is shown to further enhance the cross-lingual learning of `BilDRL`.

9.4 Experiments

We present experiments on two multilingual tasks: the cross-lingual reverse dictionary retrieval task and the bilingual paraphrase identification task.

Datasets. The experiment of cross-lingual reverse dictionary retrieval is conducted on a trilingual dataset *Wikt3l*. This dataset is extracted from Wiktionary[3], which is one of the largest freely available multilingual dictionary resources on the Web. Wikt3l contains dictionary entries of lan-

[3] https://www.wiktionary.org/

Dictionary	En-Fr	Fr-En	En-Es	Es-En
#Target words	15,666	16,857	8,004	16,986
#Definitions	50,412	58,808	20,930	56,610

Table 9.1: Statistics of the bilingual dictionary dataset Wikt3l.

guage pairs (English, French) and (English, Spanish), which form En-Fr, Fr-En, En-Es and Es-En dictionaries on four bridges of languages in total. Two types of cross-lingual definitions are extracted from Wiktionary: (i) cross-lingual definitions provided under the *Translations* sections of Wiktionary pages; (ii) monolingual definitions for words that are linked to a cross-lingual counterpart with a inter-language link[4] of Wiktionary. We exclude all the definitions of stop words in constructing the dataset, and list the statistics in Table 9.1.

Since existing datasets for paraphrase identification are merely monolingual, we contribute with another dataset *WBP3l* for cross-lingual sentential paraphrase identification. This dataset contains 6,000 pairs of bilingual sentence pairs respectively for En-Fr and En-Es settings. Within each bilingual setting, 3,000 positive cases are formed as pairs of descriptions aligned by inter-language links, which exclude the word descriptions in Wikt3l for training `BilDRL`. To generate negative examples, given a source word, we first find its 15 nearest neighbors in the embedding space. Within the nearest neighbors, we use ConceptNet (Speer et al., 2017) to filter out the synonyms of the source word, so as to prevent from generating false negative cases. Then we randomly pick one word from the filtered neighbors and pair its cross-lingual definition with the English definition of the source word to create a negative case. This process ensures that each negative case is endowed with limited dissimilarity of sentence meanings, which makes the decision more challenging. For each language setting, we randomly select 70% for training, 5% for validation, and the rest 25% for testing. Note that each language setting of this dataset thereof, matches with the quantity and partitioning of sentence pairs in the widely-used Microsoft Research Paraphrase Corpus benchmark for monolingual paraphrase identification (Das and Smith, 2009; Yin et al., 2016b). Several examples from the dataset are shown in Table 9.2.

[4] An inter-language link matches the entries of counterpart words between language versions of Wiktionary.

	Positive Examples
En:	*Being remote in space.*
Fr:	*Se trouvant à une grande distance.*
En:	*The interdisciplinary science that applies theories and methods of the physical sciences to questions of biology.*
Es:	*Ciencia que emplea y desarrolla las teorias y métodos de la física en la investigación de los sistemas biolÃşgicos.*
	Negative Examples
En:	*A person who secedes or supports secession from a political union.*
Fr:	*Contrôle politique exercé par une grande puissance sur une contrÃl'e inféodée.*
En:	*The fear of closed, tight places.*
Es:	*Pérdida o disminución considerables de la memoria.*

Table 9.2: Examples of bilingual paraphrases from WBP3l.

9.4.1 Cross-lingual Reverse Dictionary Retrieval

The objective of this task is to enable cross-lingual semantic retrieval of words based on descriptions. Besides comparing variants of `BilDRL` that adopt different sentence encoders and learning strategies, we also compare with the monolingual retrieval approach proposed by Hill et al. (Hill et al., 2016). Instead of directly associating cross-lingual word definitions, this approach learns definition-to-word mappings in a monolingual scenario. When it applies to the multilingual setting, given a lexical definition, it first retrieve the corresponding word in the source language. Then, it looks up for semantically related words in the target language using bilingual word embeddings. As discussed in Section 9.3, NMT does not apply to this task due that it cannot capture the multi-granular correspondence between a sentence and a lexeme.

Evaluation Protocol. Before training the models, we randomly select 500 defined words from each dictionary respectively as test cases, and exclude the definitions of these words from the training data. Each of the basic `BilDRL` variants are trained on one bilingual dictionary. The monolingual retrieval models are trained to fit the target words in the original languages of the word definitions, which are also provided in Wiktionary. `BilDRL` variants with multi-task or joint learning use both dictionaries of the same language pair. In the test phase, for each test case $(w^i, S_w^j) \in D_{ij}$, the prediction performs a kNN search from the definition encoding $E_{ij}(S_w^j)$, and

Languages	En-Fr			Fr-En			En-Es			Es-En		
Metric	$P@1$	$P@10$	MRR	$P@1$	$P@10$	MRR	$P@1$	$P@10$	MRR	$P@1$	$P@10$	MRR
BOW	0.8	3.4	0.011	0.4	2.2	0.006	0.4	2.4	0.007	0.4	2.6	0.007
CNN	6.0	12.4	0.070	6.4	14.8	0.072	3.8	7.2	0.045	7.0	16.8	0.088
GRU	35.6	46.0	0.380	38.8	49.8	0.410	47.8	59.0	0.496	57.6	67.2	0.604
ATT	38.8	47.4	0.411	39.8	50.2	0.425	51.6	59.2	0.534	60.4	68.4	0.629
GRU-mono	21.8	33.2	0.242	27.8	37.0	0.297	34.4	41.2	0.358	36.8	47.2	0.392
ATT-mono	22.8	33.6	0.249	27.4	39.0	0.298	34.6	42.2	0.358	39.4	48.6	0.414
GRU-MTL	43.4	49.2	0.452	44.4	52.8	0.467	50.4	60.0	0.530	63.6	71.8	0.659
ATT-MTL	46.8	56.6	0.487	47.6	56.6	0.497	55.8	62.2	0.575	66.4	75.0	0.687
ATT-joint	**63.6**	**69.4**	**0.654**	**68.2**	**75.4**	**0.706**	**69.0**	**72.8**	**0.704**	**78.6**	**83.4**	**0.803**

Table 9.3: Cross-lingual reverse dictionary retrieval results by `BilDRL` variants. We report $P@1$, $P@10$, and MRR on four groups of models: (i) basic dictionary models that adopt four different encoding techniques (BOW, CNN, GRU and ATT); (ii) models with the two best encoding techniques that enforce the monolingual retrieval approach by Hill et al. (Hill et al., 2016) (GRU-mono and ATT-mono); (iii) models adopting bilingual multi-task learning (GRU-MTL and ATT-MTL); (iv) joint learning that employs the best dictionary model of ATT-MTL (ATT-joint).

record the rank of w^i within the vocabulary of l_i. We limit the vocabularies to all words that appear in the Wikt3l dataset, which involve around 45k English words, 44k French words and 36k Spanish words. We aggregate three metrics on test cases: the accuracy $P@1$ (%), the proportion of ranks no larger than 10 $P@10$ (%), and mean reciprocal rank MRR.

We pre-train BilBOWA based on the original configuration by Gouws et al. (Gouws et al., 2015) and obtain 50-dimensional initialization of bilingual word embedding spaces respectively for the English-French and English-Spanish settings. For CNN, GRU, and attentive GRU (ATT) encoders, we stack five of each corresponding encoding layers with hidden-sizes of 200, and two affine layers are applied to the final output for dimension reduction. This encoder architecture consistently represents the best performance through our tuning. Through comprehensive hyperparameter tuning, we fix the learning rate α to 0.0005, the exponential decay rates of AMSGrad β_1 and β_2 to 0.9 and 0.999, coefficients λ_1 and λ_2 to both 0.1, and batch size to 64. Kernel-size and pooling-size are both set to 2 for CNN. Word definitions are zero-padded (short ones) or truncated (long ones) to the sequence length of 15, since most definitions (over 92%) are within 15 words in the dataset. Training is limited to 1,000 epochs for all models as well as the dictionary thread of asynchronous joint learning, in which all models are able to converge.

Results. Results are reported in Table 9.3 in four groups. The first group compares four different encoding techniques for the basic dictionary models. GRU thereof consistently outperforms CNN and BOW, since the latter two fail to capture the important sequential information for descriptions. ATT that weighs among the hidden states has notable improvements over GRU. While we equip the two better encoding techniques with the monolingual retrieval approach (GRU-mono and ATT-mono), we find that the way of learning the dictionary models towards monolingual targets and retrieving cross-lingual related words incurs more impreciseness to the task. For models of the third group that conduct multi-task learning in two directions of a language pair, the results show significant enhancement of performance in both directions. For the final group of results, we incorporate the best variant of multi-task models into the joint learning architecture, which leads to compelling improvement of the task on all settings. This demonstrates that properly adapting the word embeddings in joint with the bilingual dictionary model efficaciously constructs the embedding space that suits better the representation of both bilingual lexical and sentential semantics.

In general, this experiment has identified the proper encoding techniques of the dictionary model. The proposed strategies of multi-task and joint learning effectively contribute to the precise characterization of the cross-lingual correspondence of lexical and sentential semantics, which have led to very promising capability of cross-lingual reverse dictionary retrieval.

9.4.2 Bilingual Paraphrase Identification

The bilingual paraphrase identification problem[5] is a binary classification task with the goal to decide whether two sentences in different languages express the same meanings. `BilDRL` provides an effective solution by transferring sentential meanings to lexeme-level representations and learning a simple classifier. We evaluate three variants of `BilDRL` on this task using WBP3l: the multi-task `BilDRL` with GRU encoders (`BilDRL`-GRU-MTL), the multi-task `BilDRL` with attentive GRU encoders (`BilDRL`-ATT-MTL), and the joint learning `BilDRL` with with attentive GRU encoders (`BilDRL`-ATT-joint). We compare against several baselines of neural sentence pair

[5] Paraphrases have similar meanings, but can largely differ in content details and word orders. Hence, they are essentially different from translations. We have found that even the well-recognized Google NMT frequently caused distortions to short sentence meanings, and led to results that were close to random guess by the baseline classifiers after translation.

Languages	En&Fr		En&Es	
Metrics	*Acc.*	*F1*	*Acc.*	*F1*
BiBOW	54.93	0.622	56.27	0.623
BiCNN	54.33	0.625	53.80	0.611
ABCNN	56.73	0.644	58.83	0.655
BiLSTM	59.60	0.662	57.60	0.637
BiATT	61.47	0.699	61.27	0.689
BilDRL-GRU-MTL	64.80	0.732	63.33	0.722
BilDRL-ATT-MTL	65.27	0.735	66.07	0.735
BilDRL-ATT-joint	**68.53**	**0.785**	**67.13**	**0.759**

Table 9.4: Accuracy and F1-scores of bilingual paraphrase identification. For BilDRL, the results by three model variants are reported: BilDRL-GRU-MTL and BilDRL-ATT-MTL are models with bilingual multi-task learning, and BilDRL-ATT-joint is the best ATT-based dictionary model variant deployed with both multi-task and joint learning.

models that are proposed for monolingual paraphrase identification. These models include Siamese structures of CNN (BiCNN) (Yin and Schütze, 2015), RNN (BiLSTM) (Mueller and Thyagarajan, 2016), attentive CNN (ABCNN) (Yin et al., 2016b), attentive GRU (BiATT) (Rocktäschel et al., 2016a), and BOW (BiBOW). To support the reasoning of cross-lingual semantics, we provide the baselines with the same BilBOWA embeddings.

Evaluation protocol. BilDRL transfers each sentence into a vector in the word embedding space. Then, for each sentence pair in the train set, a Multi-layer Perceptron (MLP) with a binary softmax loss is trained on the subtraction of two vectors as a downstream classifier. Baseline models are trained end-to-end, each of which directly uses a parallel pair of encoders with shared parameters and an MLP that is stacked to the subtraction of two sentence vectors. Note that some works use concatenation (Yin and Schütze, 2015) or Manhattan distances (Mueller and Thyagarajan, 2016) of sentence vectors instead of their subtraction (Jiang et al., 2018b), which we find to be less effective on small amount of data.

We apply the configurations of the sentence encoders from the last experiment to corresponding baselines, so as to show the performance under controlled variables. Training of a classifier is terminated by early-stopping based on the validation set. Following convention (Hu et al., 2014; Yin et al., 2016b), we report the accuracy and F1 scores.

Results. This task is challenging due to the heterogeneity of cross-lingual paraphrases and limitedness of learning resources. The results in Table 9.4 show that all the baselines, where BiATT consistently outperforms the others, merely reaches slightly over 60% of accuracy on both En-Fr and En-Es settings. We believe that it comes down to the fact that sentences of different languages are often drastically heterogenous in both lexical semantics and the sentence grammar that governs the composition of lexemes. Hence, it is not surprising that previous neural sentence pair models, which capture the semantic relation of bilingual sentences directly from all participating lexemes, fall short at the multilingual task. BilDRL, however, effectively leverages the correspondence of lexical and sentential semantics to simplify the task to an easier entailment task in the lexicon space, for which the multi-task learning BilDRL-ATT-MTL outperforms the best baseline respectively by 3.80% and 4.80% of accuracy in both language settings, while BilDRL-ATT-joint, employing the joint learning, further improves the task by another satisfying 3.26% and 1.06% of accuracy. Both also show notable increment in F1.

9.5 Conclusion

In this chapter, we propose a neural embedding model BilDRL that captures the correspondence of cross-lingual lexical and sentential semantics. We experiment with multiple forms of neural models and identify the best technique. The two learning strategies, bilingual multi-task learning and joint learning, are effective at enhancing the cross-lingual learning with limited resources, and also achieve promising performance on cross-lingual reverse dictionary retrieval and bilingual paraphrase identification tasks by associating lexical and sentential semantics. An important direction of future work is to explore whether the lexeme-sentence alignment can improve bilingual word embeddings. Applying BilDRL to bilingual question answering and semantic search systems is another important direction.

CHAPTER 10

Conclusion and Future Work

10.1 Summary

In this dissertation, towards the goal of representation learning for complex multi-relational data, we make contributions on three aspects:

1. We propose a general learning framework to capture and transfer the embeddings across multiple knowledge graphs, based on simple, complex, fuzzy and insufficient alignment information.
2. We enhance the multi-relational representation learning approaches to preserve comprehensive properties of relational facts in the embedding space, including relational properties, hierarchy structures, and uncertainty.
3. We investigate generic neural sequence pair models for relational learning based on different types of sequence data, with the aim of automatically acquiring relational knowledge.

More specifically, in Chapter 3, we introduced the first method to learn transferred embeddings for different language-specific knowledge graphs. The proposed model organizes each language-specific version in a separated embedding space, and explores three representation techniques for cross-lingual knowledge transfer. This model is further extended to address the semi-supervised transfer problem based on limited supervision, by incorporating an iterative co-training process.

In Chapter 4, we extended the techniques in the previous chapter to deal with complex knowledge transfer. Two types of hierarchical grouping based alignment models are proposed to capture the many-to-one associations of entities and concepts between an instance knowledge graph and a hierarchical ontology graph in a two-view knowledge base. A Semi-non-negative Matrix Tri-factorization based alignment model is proposed to capture and propagate the fuzzy alignment information between genes and cells in single-cell RNA sequencing data.

In Chapter 5, we studied the approach to preserve relational properties of an ontology with a multi-relational embedding model. Then Chapter 6 proposed an approach to preserve both structural and uncertainty information of relation facts from an uncertain knowledge graph in the embedding space. To further enhance the precision of UKGE, we also introduce probabilistic soft logic to infer confidence scores for unseen relation facts during training.

In Chapter 7, we provided an approach to large-scale detection of the main and sub-article relations for Wikipedia articles, based on a hierarchical learning structure that combines multiple variants of neural document pair encoders with a comprehensive set of explicit features.

Chapter 8 presented an end-to-end framework to predict protein-protein interaction (PPI) knowledge using only the protein sequences. The proposed framework offers the state-of-the-art performance on predicting binary PPI, and more challenging problems of interaction type prediction and binding affinity estimation.

Finally, in Chapter 9 we presented a joint learning framework that captures the multi-granular associations of words and sentences based on lexicographic definitions.

This dissertation comprehensively extends multi-relational representation and knowledge acquisition learning techniques with the ability of knowledge transfer and the characterization of complex properties. The proposed methods benefit a wide spectrum of applications in different domains, including knowledge alignment, monolingual and cross-lingual knowledge graph completion, semantic search of entities, entity typing, paraphrase identification, uncertain relation prediction, protein-protein interaction prediction, protein binding affinity estimation, single-cell RNA-sequence imputation, and sub-article matching.

10.2 Future Directions

We outline several promising future directions expanded from the work presented in this dissertation.

1. **Capturing side information for relation representation**. This direction can be viewed from two sides. (i) The relation between two entities in some multi-relational data can contain much richer information than just a simple label. Consider that to embed an organizational chart of an engineering company, the employees not only just constitute the supervision relations from the organizational chart, but can also share side information from email logs and project collaboration records (Chen and Quirk, 2019). Another related problem is the dialogue state tracking (DST) problem for task-oriented dialogue systems (Mrkšić et al., 2017). DST can be naturally viewed as modeling the relations of dialogue states, where the relation or state-transition is captured based on system dialogues and user utterances. (ii) High-order relational dependency of multiple relation facts (Chen et al., 2018a; Hamilton et al., 2018; Tian et al., 2019) can be captured to improve the characterization of a knowledge graph, and meanwhile can help logical rule induction from the latent representations of the relations.

2. **Non-Euclidean embeddings of multi-relational data**. Another research direction extended from this work is to enable knowledge transfer across non-euclidean embedding spaces, which may include hyperbolic spaces (Nickel and Kiela, 2017), complex spaces (Trouillon et al., 2016), lie groups (Ebisu and Ichise, 2018), and perhaps Sobolev spaces (Edmunds and Rákosník, 2000). This lead to better characterization of specific structures of multi-relational data that for hierarchies or relational sequences. Transferring across such non-Enclidean spaces non-trivially requires the adaptation of norm metrics and gradient learning to conform with the corresponding non-Euclidean geometry.

3. **Multi-modal relational learning.** The sequence-based relational learning frameworks proposed in Chapters 7 to 8 may be further extended to capture the relations for objects of more

modalities, including program codes (Sivaraman et al., 2019), lineage data (Zaniolo et al., 2017), and multimedia data (Pezeshkpour et al., 2018). For such cases, different encoding techniques shall be deployed for different modalities.

4. **Transfer embeddings as background knowledge.** Another important direction is to support the knowledge transfer as background knowledge to augment label-less learning with few-shot and zero-shot cases of downstream tasks (Rios and Kavuluru, 2018; Zhou et al., 2017), or improve deep learning based tasks with label reduction and partial label cases (Zhang and Yu, 2015).

Bibliography

Ackerman, M. S., Dachtera, J., et al. (2013). Sharing knowledge and expertise: the cscw view of knowledge management. *CSCW*.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *NAR*, 25(17):3389–3402.

Anderson, C. (2018). Google's ai tool deepvariant promises significantly fewer genome errors. *Clinical OMICs*, 5(1):33–33.

Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294.

Artetxe, M., Labaka, G., and Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

Bach, S., Huang, B., London, B., and Getoor, L. (2013). Hinge-loss markov random fields: Convex inference for structured prediction.

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. (2017). Hinge-loss markov random fields and probabilistic soft logic. *JMLR*.

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics.

Bard, J., Rhee, S. Y., et al. (2005). An ontology for cell types. *Genome biology*.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1).

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *NAR*, 28(1):235–242.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Bond, F. and Foster, R. (2013). Linking and extending an open multilingual Wordnet. In *ACL*.

Bordes, A., Glorot, X., et al. (2012). Joint learning of words and meaning representations for open-text semantic parsing. In *AISTATS*.

Bordes, A., Glorot, X., et al. (2014a). A semantic matching energy function for learning with multi-relational data. *Machine Learning*.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *NIPS*.

Bordes, A., Weston, J., et al. (2011). Learning structured embeddings of knowledge bases. In *AAAI*.

Bordes, A., Weston, J., and Usunier, N. (2014b). Open question answering with weakly supervised embedding models. In *ECML-PKDD*. Springer.

Cai, Z., Zhao, K., et al. (2013). Wikification via link co-occurrence. In *CIKM*.

Camossi, E., Bertolotto, M., et al. (2006). A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *IJGIS*.

Chandar, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861.

Chen, D., Fisch, A., et al. (2017a). Reading Wikipedia to answer open-domain questions. In *ACL*.

Chen, H., Sun, X., Tian, Y., et al. (2018a). Enhanced network embeddings via exploiting edge labels. In *CIKM*.

Chen, J., Chen, J., and Yu, Z. (2019a). Incorporating structured commonsense knowledge in story completion. In *AAAI*.

Chen, M., Gao, S., et al. (2016a). Converting spatiotemporal data among heterogeneous granularity systems. In *FUZZ-IEEE*.

Chen, M., Gao, S., et al. (2016b). Converting spatiotemporal data among multiple granularity systems. In *SAC*.

Chen, M., Ju, C., Zhou, G., Chen, X., Zhang, T., Chang, K.-W., Zaniolo, C., and Wang, W. (2019b). Multifaceted protein-protein interaction prediction based on siamese residual rcnn. *Bioinformatics (to appear)*.

Chen, M., Meng, C. P., Huang, G., and Zaniolo, C. (2018b). Neural article pair modeling for wikipedia sub-article machine. In *ECML-PKDD*.

Chen, M. and Quirk, C. (2019). Embedding edge-attributed relational hierarchies. In *SIGIR*.

Chen, M., Tian, Y., Chang, K.-W., Skiena, S., and Zaniolo, C. (2018c). Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI*.

Chen, M., Tian, Y., Chen, X., Xue, Z., and Zaniolo, C. (2018d). On2vec: Embedding-based relation prediction for ontology population. In *SDM*.

Chen, M., Tian, Y., et al. (2017b). Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *IJCAI*.

Chen, M., Tian, Y., et al. (2017c). Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*.

Chen, M., Tian, Y., et al. (2018e). Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment. In *IJCAI*.

Chen, M., Tian, Y., et al. (2018f). On2vec: Embedding-based relation prediction for ontology population. In *SDM*.

Chen, M., Weinberger, K. Q., Xu, Z., and Sha, F. (2015). Marginalizing stacked linear denoising autoencoders. *JMLR*.

Chen, M. and Zaniolo, C. (2017). Learning multi-faceted knowledge graph embeddings for natural language processing. *IJCAI*.

Chen, M., Zhou, T., et al. (2017d). Multi-graph affinity embeddings for multilingual knowledge graphs. In *AKBC*.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *KDD*.

Chen, T., Sun, Y., et al. (2017e). On sampling strategies for neural network-based collaborative filtering. In *KDD*.

Chen, X., Chen, M., Shi, W., Sun, Y., and Zaniolo, C. (2019c). Embedding uncertain knowledge graphs. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*.

Cheng, J., Sun, S., et al. (2004). Netaffx gene ontology mining tool: a visual approach for microarray data analysis. *Bioinformatics*, 20(9).

Chin, W.-S., Yuan, B.-W., Yang, M.-Y., Zhuang, Y., Juan, Y.-C., and Lin, C.-J. (2016). Libmf: a library for parallel matrix factorization in shared-memory systems. *JMLR*, 17(1):2971–2975.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Chopra, S., Auli, M., and Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Cilibrasi, R. L. and Vitanyi, P. M. (2007). The google similarity distance. *TKDE*, 19(3).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017a). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. (2017b). Very deep convolutional networks for text classification. In *EACL*.

Consortium, U. et al. (2018). Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 46(5):2699.

Coulmance, J., Marty, J.-M., Wenzek, G., and Benhalloum, A. (2015). Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113.

Culotta, A. and Sorensen, J. (2004). Dependency tree kernels for relation extraction. In *ACL*.

Das, D. and Smith, N. A. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, pages 468–476. Association for Computational Linguistics.

Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., Smola, A., and McCallum, A. (2018). Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *ICLR*.

Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Comm ACM*, 51(1).

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *AAAI*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1370–1380.

Dhingra, B., Liu, H., et al. (2016). Gated-attention readers for text comprehension. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Ding, C. H., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *TPAMI*, 32(1):45–55.

Dojchinovski, M. and Kliegr, T. (2013). Entityclassifier. eu: real-time classification of entities in text with wikipedia. In *ECML-PKDD*.

Doval, Y., Camacho-Collados, J., Anke, L. E., and Schockaert, S. (2018). Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 294–304.

Du, X., Sun, S., Hu, C., Yao, Y., Yan, Y., and Zhang, Y. (2017). Deepppi: boosting prediction of protein–protein interactions with deep neural networks. *JCIM*, 57(6):1499–1510.

Durif, G., Modolo, L., Mold, J., Lambert-Lacroix, S., and Picard, F. (2017). Probabilistic count matrix factorization for single cell expression data analysis. In *RECOMB*, page 254. Springer.

Ebisu, T. and Ichise, R. (2018). Toruse: Knowledge graph embedding on a lie group. In *AAAI*.

Edmunds, D. and Rákosník, J. (2000). Sobolev embeddings with variable exponent. *Studia Mathematica*, 3(143):267–293.

Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., and Riedel, S. (2016). emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Elyanow, R., Dumitrascu, B., Engelhardt, B. E., and Raphael, B. J. (2019). netnmf-sc: Leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis. In *RECOMB*.

Fang, Y., Kuan, K., Lin, J., Tan, C., and Chandrasekhar, V. (2017). Object detection meets knowledge graphs.

Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. *EACL*.

Féraud, R. and Clérot, F. (2002). A methodology to explain neural network classification. *Neural Networks*, 15(2).

Fields, S. and Song, O.-k. (1989). A novel genetic system to detect protein–protein interactions. *Nature*, 340(6230):245.

Fundel, K., Küffner, R., et al. (2007). RelEx–Relation extraction using dependency parse trees. *Bioinformatics*, 23(3).

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*.

Gavin, A.-C., Bösche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J. M., Michon, A.-M., Cruciat, C.-M., et al. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141.

Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N., and Garry, D. J. (2018). Drimpute: imputing dropout events in single cell rna sequencing data. *BMC bioinformatics*, 19(1):220.

Gouws, S., Bengio, Y., and Corrado, G. (2015). Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 748–756.

Guo, Y., Yu, L., Wen, Z., and Li, M. (2008). Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *NAR*, 36(9):3025–30.

Hamilton, W., Bajaj, P., Zitnik, M., Jurafsky, D., and Leskovec, J. (2018). Embedding logical queries on knowledge graphs. In *NIPS*, pages 2026–2037.

Hao, J., Chen, M., Yu, W., Sun, Y., and Wang, W. (2019). Universal representationlearning of knowledge bases by jointly embedding instances and ontological concepts. In *KDD*.

Hashemifar, S., Neyshabur, B., Khan, A. A., and Xu, J. (2018). Predicting protein–protein interactions through sequence-based deep learning. *Bioinformatics*, 34(17).

He, H., Balakrishnan, A., Eric, M., and Liang, P. (2017a). Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *ACL*.

He, H., Gimpel, K., and Lin, J. (2015). Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*, pages 1576–1586.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*, pages 770–778.

He, R., Kang, W.-C., and McAuley, J. (2017b). Translation-based recommendation. In *Recsys*, pages 161–169. ACM.

Hecht, B., Carton, S. H., et al. (2012). Explanatory semantic relatedness and explicit spatialization for exploratory search. In *SIGIR*.

Hill, F., Cho, K., Korhonen, A., and Bengio, Y. (2016). Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S.-L., Millar, A., Taylor, P., Bennett, K., Boutilier, K., et al. (2002). Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180.

Hu, B., Lu, Z., Li, H., and Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Hu, J., Cheng, R., Huang, Z., Fang, Y., and Luo, S. (2017). On embedding uncertain graphs. In *CIKM*.

Huang, M., Wang, J., Torre, E., Dueck, H., Shaffer, S., Bonasio, R., Murray, J. I., Raj, A., Li, M., and Zhang, N. R. (2018). Saver: gene expression recovery for single-cell rna sequencing. *Nature methods*, 15(7):539.

Huang, X., Zhang, J., Li, D., and Li, P. (2019). Knowledge graph embedding based question answering. In *WSDM*, pages 105–113. ACM.

Huang, Y.-A., You, Z.-H., Gao, X., Wong, L., and Wang, L. (2015). Using weighted sparse representation model combined with discrete cosine transformation to predict protein-protein interactions from protein sequence. *BioMed research intl.*, 2015.

Hulstijn, J. H., Hollander, M., and Greidanus, T. (1996). Incidental vocabulary learning by advanced foreign language students: The influence of marginal glosses, dictionary use, and reoccurrence of unknown words. *The modern language journal*, 80(3):327–339.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., and Gerstein, M. (2003). A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302:449–53.

Jenatton, R., Roux, N. L., Bordes, A., and Obozinski, G. R. (2012). A latent factor model for highly multi-relational data. In *NIPS*.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In *ACL*.

Ji, G., Liu, K., He, S., Zhao, J., et al. (2017). Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the AAAI International Conference on Artificial Intelligence*, pages 3060–3066.

Jia, Y., Wang, Y., Lin, H., Jin, X., and Cheng, X. (2016). Locally adaptive translation for knowledge graph embedding. In *AAAI*, pages 992–998.

Jiang, J.-Y., Chen, F., Chen, Y.-Y., and Wang, W. (2018a). Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *NAACL*.

Jiang, J.-Y., Chen, F., Chen, Y.-Y., and Wang, W. (2018b). Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1812–1822.

Kadlec, R., Schmid, M., et al. (2016). Text understanding with the attention sum reader network. In *ACL*, volume 1.

Kalchbrenner, N., Grefenstette, E., Blunsom, P., Kartsaklis, D., Kalchbrenner, N., Sadrzadeh, M., Kalchbrenner, N., Blunsom, P., Kalchbrenner, N., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 212–217. Association for Computational Linguistics.

Kim, J., Kwon Lee, J., and Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP*.

Kimmig, A., Bach, S., Broecheler, M., Huang, B., and Getoor, L. (2012). A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Kittur, A. and Kraut, R. E. (2010). Beyond wikipedia: coordination and conflict in online production groups. In *CSCW*.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Kotnis, B. and Nastase, V. (2017). Analysis of the impact of negative sampling on link prediction in knowledge graphs. *arXiv preprint arXiv:1708.06816*.

Lascarides, A. and Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and philosophy*, 16(5).

Lau, R. Y., Song, D., et al. (2009). Toward a fuzzy domain ontology extraction method for adaptive e-learning. *TKDE*, 21(6).

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Lehmann, J., Isele, R., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2).

Levine, J. H., Simonds, E. F., Bendall, S. C., Davis, K. L., El-ad, D. A., Tadmor, M. D., Litvin, O., Fienberg, H. G., Jager, A., Zunder, E. R., et al. (2015). Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197.

Li, H., Gong, X.-J., Yu, H., and Zhou, C. (2018). Deep neural network based predictions of protein interactions using primary sequences. *Molecules*, 23(8):1923.

Li, H., Liu, T.-Y., and Zhai, C. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*.

Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659.

Li, W. V. and Li, J. J. (2018). An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications*, 9(1):997.

Lin, C.-Y. and Hovy, E. (2002). From single to multi-document summarization: A prototype system and its evaluation. In *ACL*.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. In *ICLR*.

Lin, P., Troup, M., and Ho, J. W. (2017a). Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, 18(1):59.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *AAAI*.

Lin, Y., Shen, S., et al. (2016). Neural relation extraction with selective attention over instances. In *ACL*.

Lin, Y., Yu, B., et al. (2017b). Problematizing and addressing the article-as-concept assumption in wikipedia. In *CSCW*.

Link, A. (2019). `https://www.dropbox.com/sh/jcjsy0s2ozvhx6x/AACSxhUC8CG1l3kRFQeHwrPOa?dl=0`.

Liu, X., Xia, T., et al. (2016). Cross social media recommendation. In *ICWSM*.

Lukasiewicz, T. and Straccia, U. (2008). Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Luong, T., Pham, H., and Manning, C. D. (2015a). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159.

Luong, T., Pham, H., and Manning, C. D. (2015b). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Lv, X., Hou, L., Li, J., and Liu, Z. (2018). Differentiating concepts and instances for knowledge graph embedding. In *EMNLP*.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, page 3.

Mahdisoltani, F., Biega, J., et al. (2015). Yago3: A knowledge base from multilingual Wikipedias. In *CIDR*.

McLachlan, G., Do, K.-A., and Ambroise, C. (2005). *Analyzing microarray gene expression data*, volume 422. John Wiley & Sons.

Meij, E., Balog, K., and Odijk, D. (2014). Entity linking and retrieval for semantic search. In *WSDM*.

Meyer, C. M. and Gurevych, I. (2012). *Wiktionary: A new rival for expert-built lexicons? Exploring the possibilities of collaborative lexicography*. na.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013a). Exploiting similarities among languages for machine translation.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Mikolov, T., Sutskever, I., et al. (2013c). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In *CIKM*.

Min, X., Zeng, W., Chen, N., Chen, T., and Jiang, R. (2017). Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding. *Bioinformatics*, 33(14):i92–i101.

Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al. (2018). Never-ending learning. *Communications of the ACM*.

Moal, I. H. and Fernández-Recio, J. (2012). Skempi: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. *Bioinformatics*, 28(20):2600–2607.

Mousavi, H., Atzori, M., et al. (2014a). Text-mining, structured queries, and knowledge management on web document corpora. *SIGMOD Record*.

Mousavi, H., Gao, S., , et al. (2014b). Mining semantics structures from syntactic structures in web document corpora. *International Journal of Semantic Computing*, 8(04).

Mrkšić, N., Séaghdha, D. Ó., Wen, T.-H., Thomson, B., and Young, S. (2017). Neural belief tracker: Data-driven dialogue state tracking. In *ACL*, pages 1777–1788.

Mueller, J. and Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2786–2792.

Müller, A. T., Hiss, J. A., and Schneider, G. (2018). Recurrent neural network model for constructive peptide design. *JCIM*, 58(2):472–479.

Needell, D., Ward, R., et al. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *NIPS*.

Nenkova, A., McKeown, K., et al. (2012). A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer.

Nguyen, D. Q., Sirts, K., et al. (2016). Stranse: a novel embedding model of entities and relationships in knowledge bases. In *NAACL*.

Nguyen, T., Moreira, V., et al. (2011). Multilingual schema matching for Wikipedia infoboxes. *PVLDB*.

Ni, Y., Xu, Q. K., et al. (2016). Semantic documents relatedness using concept graph representation. In *WSDM*, pages 635–644.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *NIPS*.

Nickel, M., Rosasco, L., Poggio, T. A., et al. (2016). Holographic embeddings of knowledge graphs. In *AAAI*.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *ICML*.

Olden, J. D. and Jackson, D. A. (2002). Illuminating the ÂąÂřblack boxÂąÂś: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1-2).

Pan, X. and Shen, H.-B. (2018). Predicting rna–protein binding sites and motifs through combining local and global deep convolutional neural networks. *Bioinformatics*, 34(20).

Pan, X.-Y., Zhang, Y.-N., and Shen, H.-B. (2010). Large-scale prediction of human protein- protein interactions from amino acid sequence based on latent topic features. *Journal of Proteome Research*, 9(10):4992–5001.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Petta, I., Lievens, S., Libert, C., Tavernier, J., and De Bosscher, K. (2016). Modulation of protein–protein interactions for the development of novel therapeutics. *Molecular Therapy*, 24(4):707–718.

Pezeshkpour, P., Chen, L., and Singh, S. (2018). Embedding multimodal relational data for knowledge base completion. In *EMNLP*, pages 3208–3218.

Philipp, O., Osiewacz, H. D., and Koch, I. (2016). Path2ppi: an r package to predict protein–protein interaction networks for a set of proteins. *Bioinformatics*, 32(9).

Poria, S., Cambria, E., et al. (2015). Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *EMNLP*.

Pujara, J., Augustine, E., and Getoor, L. (2017). Sparsity and noise: Where knowledge graph embeddings fall short. In *EMNLP*, pages 1751–1756.

Quan, T. T., Hui, S. C., et al. (2004). Automatic generation of ontology for scholarly semantic web. In *ISWC*.

Quang, D. and Xie, X. (2016). Danq: a hybrid convolutional and recurrent deep neural network for quantifying the function of dna sequences. *NAR*, 44(11).

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *CoNLL*.

Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations*.

Rinser, D., Lange, D., et al. (2013). Cross-lingual entity matching and infobox alignment in Wikipedia. *Information Systems*, 38(6).

Rios, A. and Kavuluru, R. (2018). Few-shot and zero-shot multi-label learning for structured label spaces. In *EMNLP*, volume 2018, page 3132.

Ristoski, P., Rosati, J., Di Noia, T., De Leone, R., and Paulheim, H. (2018). Rdf2vec: Rdf graph embeddings and their applications. *Semantic Web*, pages 1–32.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. (2016a). Reasoning about entailment with neural attention. In *International Conference on Learning Representations*.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kocisky, T., and Blunsom, P. (2016b). Reasoning about entailment with neural attention. In *ICLR*.

Ruder, S., Vulić, I., and Søgaard, A. (2017). A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*.

Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., and Eisenberg, D. (2004). The database of interacting proteins: 2004 update. *NAR*, 32(suppl_1):D449–D451.

Saxe, A. M., McClelland, J. L., et al. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *ICLR*.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Schuhmacher, M. and Ponzetto, S. P. (2014). Knowledge-based graph document modeling. In *WSDM*.

Scott, D. E., Bayly, A. R., Abell, C., and Skidmore, J. (2016). Small molecules, big targets: drug discovery faces the protein–protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8):533.

Sha, L., Chang, B., et al. (2016). Reading and thinking: Re-read lstm unit for textual entailment recognition. In *Proceedings of the International Conference on Computational Linguistics*.

Shekhar, K., Lapan, S. W., Whitney, I. E., Tran, N. M., Macosko, E. Z., Kowalczyk, M., Adiconis, X., Levin, J. Z., Nemesh, J., Goldman, M., et al. (2016). Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323.

Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y., and Jiang, H. (2007). Predicting protein-protein interactions based only on sequences information. *PNAS*, 104(11):4337–41.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Shi, B. and Weninger, T. (2017). Proje: Embedding projection for knowledge graph completion. In *AAAI*.

Silberberg, Y., Kupiec, M., and Sharan, R. (2014). A method for predicting protein-protein interaction types. *PLoS One*, 9(3).

Sivaraman, A., Zhang, T., Broeck, G. V. d., and Kim, M. (2019). Active inductive logic programming for code search. In *ICSE*.

Skrabanek, L., Saini, H. K., Bader, G. D., and Enright, A. J. (2008). Computational prediction of protein–protein interactions. *Molecular biotechnology*, 38(1):1–17.

Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

Speer, R. and Havasi, C. (2013). Conceptnet 5: A large semantic network for relational knowledge. *The People's Web Meets NLP*, pages 161–176.

Srinivasulu, Y. S., Wang, J.-R., Hsu, K.-T., Tsai, M.-J., Charoenkwan, P., Huang, W.-L., Huang, H.-L., and Ho, S.-Y. (2015). Characterizing informative sequence descriptors and predicting binding affinities of heterodimeric protein complexes. *BMC bioinformatics*, 16(18):S14.

Strube, M. and Ponzetto, S. P. (2006). Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*.

Suchanek, F. M., Abiteboul, S., et al. (2011). Paris: Probabilistic alignment of relations, instances, and schema. *PVLDB*, 5(3).

Sun, T., Zhou, B., Lai, L., and Pei, J. (2017a). Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics*, 18(1).

Sun, Z., Hu, W., and Li, C. (2017b). Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*.

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., Santos, A., Doncheva, N. T., Roth, A., Bork, P., et al. (2016). The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic acids research*.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, pages 1556–1566.

Talwar, D., Mongia, A., Sengupta, D., and Majumdar, A. (2018). Autoimpute: Autoencoder based imputation of single-cell rna-seq data. *Scientific reports*, 8(1):16329.

Thi, D., Quynh, N., et al. (2016). Facing the most difficult case of semantic role labeling: A collaboration of word embeddings and co-training. In *ACL*.

Tian, Y., Chen, H., Perozzi, B., Chen, M., Sun, X., and Skiena, S. (2019). Social relation inference via label propagation. In *ECIR*.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *ICML*.

Tsai, C.-T. and Roth, D. (2016). Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598.

Upadhyay, S., Gupta, N., and Roth, D. (2018). Joint multilingual supervision for cross-lingual entity linking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2495.

van Dijk, D., Nainys, J., Sharma, R., Kathail, P., Carr, A. J., Moon, K. R., Mazutis, L., Wolf, G., Krishnaswamy, S., and Pe'er, D. (2017). Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data. *BioRxiv*, page 111591.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Vrandečić, D. (2012). Wikidata: A new platform for collaborative data collection. In *WWW*.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledge base. *CACM*.

Vulić, I., Korhonen, A., et al. (2016). On the role of seed lexicons in learning bilingual word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 247–257.

Vulić, I. and Moens, M.-F. (2015). Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 363–372. ACM.

Vyas, Y. and Carpuat, M. (2016). Sparse bilingual word representations for cross-lingual lexical entailment. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1187–1197.

Wan, X. (2009). Co-training for cross-lingual sentiment classification. In *ACL-IJCNLP*.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017a). Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE*.

Wang, Y.-B., You, Z.-H., Li, X., Jiang, T.-H., Chen, X., Zhou, X., and Wang, L. (2017b). Predicting protein–protein interactions from protein sequences by a stacked sparse autoencoder deep neural network. *Molecular BioSystems*, 13(7):1336–1344.

Wang, Z., Gerstein, M., and Snyder, M. (2009). Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57.

Wang, Z., Li, J., et al. (2012). Cross-lingual knowledge linking across wiki knowledge bases. In *WWW*.

Wang, Z. and Wang, H. (2016). Understanding short texts. In *ACL*.

Wang, Z., Wang, H., Wen, J.-R., and Xiao, Y. (2015). An inference approach to basic level of categorization. In *CIKM*.

Wang, Z., Zhang, J., et al. (2014a). Knowledge graph and text jointly embedding. In *EMNLP*.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014b). Knowledge graph embedding by translating on hyperplanes. In *AAAI*.

Welch, B. L. (1947). The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35.

Weston, J., Bordes, A., et al. (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *EMNLP*.

Widyantoro, D. H. and Yen, J. (2001). A fuzzy ontology-based abstract search engine and its user studies. In *FUZZ-IEEE*.

Wikipedia (2016). https://www.wikipedia.org/.

Wilson, D. R. and Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10).

Wong, L., You, Z.-H., Li, S., Huang, Y.-A., and Liu, G. (2015). Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel pr-lpq descriptor. In *ICIC*, pages 713–720.

Wu, W., Li, H., Wang, H., and Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. In *AAAI*, pages 2659–2665.

Xing, C., Wang, D., Liu, C., and Lin, Y. (2015). Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011.

Yamada, I., Shindo, H., et al. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *CoNLL*.

Yang, B., Yih, W.-t., et al. (2015a). Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2015b). Embedding entities and relations for learning and inference in knowledge bases. *ICLR*.

Yang, C., Liu, Z., et al. (2015c). Network representation learning with rich text information. In *IJCAI*.

Yang, L., Xia, J.-F., and Gui, J. (2010). Prediction of protein-protein interactions from protein sequence using local descriptors. *Protein and Peptide Letters*, 17(9):1085–1090.

Yang, Y., Sun, Y., et al. (2015d). Entity matching across heterogeneous sources. In *KDD*.

Yih, W.-t., Chang, M.-W., Meek, C., and Pastusiak, A. (2013). Question answering using enhanced lexical semantic models. In *ACL*.

Yin, W. and Schütze, H. (2015). Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yin, W., Schütze, H., et al. (2016a). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4(1).

Yin, W., Schütze, H., et al. (2016b). Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4(1).

You, Z.-H., Chan, K. C., and Hu, P. (2015). Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. *PLoS One*, 10(5).

You, Z.-H., Lei, Y.-K., Zhu, L., Xia, J., and Wang, B. (2013). Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *BMC bioinformatics*, 14.

You, Z.-H., Zhu, L., Zheng, C.-H., Yu, H.-J., Deng, S.-P., and Ji, Z. (2014). Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set. *BMC bioinformatics*, 15:S9.

Yu, W., Gwinn, M., Dotson, W. D., Green, R. F., Clyne, M., Wulf, A., Bowen, S., Kolor, K., and Khoury, M. J. (2016). A knowledge base for tracking the impact of genomics on population health. *Genetics in Medicine*, 18(12):1312.

Yugandhar, K. and Gromiha, M. M. (2014). Protein–protein binding affinity prediction from amino acid sequence. *Bioinformatics*, 30(24).

Zaniolo, C., Gao, S., Atzori, M., Chen, M., and Gu, J. (2017). User-friendly temporal queries on historical knowledge bases. *Information and Computation*.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., Marques, S., Munguba, H., He, L., Betsholtz, C., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142.

Zeng, D., Liu, K., et al. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.

Zhang, M., Tang, J., et al. (2014). Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *SIGIR*.

Zhang, M.-L. and Yu, F. (2015). Solving the partial label learning problem: An instance-based approach. In *IJCAI*.

Zhang, S., Zhou, J., Hu, H., Gong, H., Chen, L., Cheng, C., and Zeng, J. (2015). A deep learning framework for modeling structural features of rna-binding protein targets. *NAR*, 44(4):e32–e32.

Zhang, Y., Chan, W., and Jaitly, N. (2017). Very deep convolutional networks for end-to-end speech recognition. In *ICASSP*, pages 4845–4849.

Zhong, H., Zhang, J., et al. (2015). Aligning knowledge and text embeddings by entity descriptions. In *EMNLP*.

Zhou, T., Chen, M., Yu, J., and Terzopoulos, D. (2017). Attention-based natural language person retrieval. In *CVPR*, pages 27–34.

Zhou, X., Wan, X., and Xiao, J. (2016). Cross-lingual sentiment classification with bilingual document representation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1403–1412.

Zhu, H., Domingues, F. S., Sommer, I., and Lengauer, T. (2006). Noxclass: prediction of protein-protein interaction types. *BMC bioinformatics*, 7(1):27.

Zhu, H., Xie, R., et al. (2017). Iterative entity alignment via knowledge embeddings. In *IJCAI*.

Zhu, J., Zou, H., Rosset, S., and Hastie, T. (2009). Multi-class adaboost. *Statistics and its Interface*, 2(3).

Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.

Zou, L., Huang, R., et al. (2014). Natural language question answering over rdf: a graph data driven approach. In *SIGMOD*.