

UCLA

UCLA Electronic Theses and Dissertations

Title

Incorporating Ontological Information in Knowledge Graph Learning and Empowered Interdisciplinary Applications

Permalink

<https://escholarship.org/uc/item/6w38395m>

Author

Hao, Junheng

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Incorporating Ontological Information in Knowledge Graph Learning and Empowered
Interdisciplinary Applications

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Junheng Hao

2022

© Copyright by
Junheng Hao
2022

ABSTRACT OF THE DISSERTATION

Incorporating Ontological Information in Knowledge Graph Learning and Empowered Interdisciplinary Applications

by

Junheng Hao

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2022

Professor Wei Wang, Co-chair

Professor Yizhou Sun, Co-chair

Large-scale Knowledge Graphs (KGs), such as Wikipedia and many enterprises or other domain-specific KGs, contain large numbers of real-world facts and are ubiquitous and foundational to many downstream knowledge-driven AI applications. Many existing techniques have applied state-of-the-art machine learning (ML) techniques in knowledge graph modeling to improve the performance in these applications with the KG backend, but the semantic structures especially the hierarchical ontological information inside the KGs are sparsely investigated and therefore relatively less leveraged into KG learning.

In this dissertation, we demonstrate a series of research results that systematically explores how such hierarchical ontological components in knowledge graphs are incorporated into KG representation learning. We present multiple practical machine learning methods, such as hierarchical graph modeling, graph neural networks, self-supervised learning and language models, that can effectively and efficiently capture ontological information, given different knowledge graph formulations. As a result, our proposed approaches address various real-world challenges in multiple domains, from knowledge graph itself, to diverse disciplines including natural language processing, recommender system, even bioinformatics and societal studies, and expand ML frontiers to knowledge graphs.

The dissertation of Junheng Hao is approved.

Kai-Wei Chang

Ying-Nian Wu

Wei Wang, Committee Co-chair

Yizhou Sun, Committee Co-chair

University of California, Los Angeles

2022

To my family, and myself.
A journey to end and to start.

TABLE OF CONTENTS

List of Figures	xi
List of Tables	xvi
Acknowledgments	xix
Vita	xxii
1 Introduction	1
1.1 Knowledge Bases and Graphs	1
1.2 Machine Learning on Knowledge Graphs	3
1.3 Knowledge Graph Empowered Applications	4
1.4 Research Roadmap and Thesis Contribution	5
1.5 Thesis Overview	6
2 Universal Representation Learning of Knowledge Bases by Jointly Embed- ding Instances and Ontological Concepts	10
2.1 Introduction	10
2.2 Related Work	13
2.3 Modeling	15
2.3.1 Formalization of Knowledge Bases	15
2.3.2 Cross-view Association Model	16
2.3.3 Intra-view Model	18
2.3.4 Joint Training on Two-View KBs	21
2.3.5 Variants of JOIE and Complexity	22

2.4	Experiments	23
2.4.1	Datasets	23
2.4.2	KG Triple Completion	24
2.4.3	Entity Typing	27
2.4.4	Case Study	30
2.4.5	Ablation Study	34
2.5	Conclusion	38
3	Bio-JOIE: Joint Representation Learning of Biological Knowledge Bases	40
3.1	Introduction	40
3.2	Related Work	44
3.3	Materials and Method	45
3.3.1	Preliminary	46
3.3.2	Knowledge Model	47
3.3.3	Transfer Model	49
3.3.4	Joint Learning Objectives	52
3.4	Results	53
3.4.1	Dataset	53
3.4.2	Baselines	55
3.4.3	PPI Type Prediction on Multiple Species	56
3.4.4	Identifying Protein Families And Enzyme Commission Based Clustering	60
3.4.5	Case Study: SARS-CoV-2-Human Protein Target Prediction	61
3.5	Extension: Bio-JOIE Inference on Texera (Collaborative Machine Learning Demonstration)	66

3.6	Conclusion	68
4	P-Companion: A Product Graph Based Principled Framework for Diversified Complementary Product Recommendation	71
4.1	Introduction	71
4.2	Preliminaries	76
4.2.1	Behavior-based Product Graph (BPG)	76
4.2.2	From two-view KG to two-view PKG	77
4.2.3	Data analysis in BPG	77
4.2.4	Problem Definition: Complementary Recommendation	80
4.3	Modeling	80
4.3.1	Product2vec: Pretrained Product Representation	82
4.3.2	Complementary Type Transition	84
4.3.3	Complementary Item Prediction	85
4.3.4	Joint Training	86
4.3.5	Training and Inference	87
4.4	Experiments	88
4.4.1	Experiment Setup	89
4.4.2	Historical Co-Purchase Evaluation	90
4.4.3	MTurk Evaluation	93
4.4.4	Production Deployment	97
4.4.5	Case Study	98
4.5	Related Work	100
4.6	Conclusion	103

5	MEDTO: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks	104
5.1	Introduction	104
5.2	Preliminaries and System Overview	108
5.2.1	Graph Neural Networks	108
5.2.2	Problem Formulation	109
5.2.3	System Overview	109
5.3	Ontology Bootstrapping from Medical Database	110
5.3.1	Ontology Creation	110
5.3.2	Ontology Enrichment	111
5.4	Ontology Matching	113
5.4.1	Input Embeddings of Medical Concepts	113
5.4.2	Hyperbolic Graph Convolution Layer	114
5.4.3	Heterogeneous Graph Module	116
5.4.4	Matching Module	117
5.4.5	Training	118
5.5	Experiments	118
5.5.1	Datasets	118
5.5.2	Compared Methods	119
5.5.3	Implementation Details	120
5.5.4	Experimental Results	120
5.6	Related Work	125
5.7	Conclusion	127
6	Multi-source Knowledge Graph Transfer	128

6.1	Introduction	128
6.2	Problem Statement	131
6.3	Methodology	132
6.3.1	Intra-Graph Encoder	133
6.3.2	Attention-based Cross-graph Transfer	134
6.3.3	Graph Decoder	136
6.3.4	Training, Inference and Complexity	136
6.4	Experiments	137
6.4.1	Datasets	137
6.4.2	Baseline Methods	138
6.4.3	Experiment Setup	140
6.4.4	Results	141
6.4.5	Hyperparameters	142
6.5	Related Work	145
6.6	Conclusion and Future work	147
7	Empowering Homicide Analytics with MurderBook Knowledge Graphs and Domain-specific Language Models	148
7.1	Introduction	148
7.2	Preliminaries	150
7.3	Methodology	152
7.3.1	MKG: Construction	153
7.3.2	M-BERT: “Crime” Language Model	154
7.3.3	EIHA: KG-infused Representation Learning Framework	155
7.4	Experiment: Case Classification	157

7.4.1	Dataset: Source and Statistics	157
7.4.2	Experimental Setup	159
7.4.3	Results	160
7.5	Applications	161
7.6	Related Work	163
7.7	Conclusion and Future Directions	164
7.8	Limitations	165
7.9	Ethical Considerations and Broader Impact	165
8	Conclusion	169

LIST OF FIGURES

1.1	Searching results of “Pablo Alborán” returned by Google, as of Aug 31, 2022.	2
1.2	Examples of multidisciplinary applications with KG backends.	5
1.3	Research roadmap of this dissertation in the format of one knowledge graph, starting from bottom left JOIE node. White, red, blue and purple nodes represent main projects covered in this dissertation, applications/tasks, application domains, and proposed techniques/technical components respectively.	6
2.1	An example of two-view KB. Regular meta-relations and hierarchical meta-relations are denoted as orange and black dashed lines respectively in the ontology view.	11
2.2	JOIE learns two aspects of a KB. The cross-view association model learns embeddings from cross-view links (dash arrows in green “category” box). The default intra-view model learns embeddings from triples (grey box) in each view; Besides, hierarchy-aware intra-view models the meta-relation facts that form hierarchies in the ontology (orange “Hierarchy” trapezoid).	15
2.3	Intuition of the cross-view association model: Cross-view Grouping (a); Cross-view Transformation (b).	17
2.4	Intuition on how the two-view KG can help with ontology population, i.e. to infer potential new relations between concepts.	30
2.5	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.	31
2.6	Long-tail distribution holds on entity frequency from both YAGO26K-906(a) and DB111K-174(b)	33
2.7	Performances of entity typing task on both datasets with different entity and concept embedding dimensionalities	36

2.8	The effect of training the model using different proportions of cross-view links on (a) YAGO26K-906 and (b) DB111K-174	37
2.9	Visualize effects on embeddings of negative sampling on cross-view links	38
3.1	Two examples of SARS-CoV-2-human protein interactions: M protein (left) and ORF3a protein (right). The purple diamonds refer to the viral proteins and the orange circles refer to the high-confidence human protein target. Proteins highlighted in blue are involved in certain biological processes, and proteins highlighted in yellow are arranged in a protein complex.	41
3.2	Examples of gene ontology annotation enrichment on three representative SARS-CoV or SARS-CoV-2 proteins, which possess multiple properties across three biological aspects: biological processes, cellular components and molecular functions.	42
3.3	Model architecture of Bio-JOIE. The Knowledge Model seeks to encode relational facts in each domain respectively (such as proteins and gene ontology). Meanwhile, the Transfer Model learns to connect both domains and enable knowledge transfer across protein and gene ontology.	48
3.4	Explanation of weighted transfer model for modeling hierarchical gene ontology.	51
3.5	Different scopes of input to train Bio-JOIE for SARS-CoV-2 PPI prediction.	62
3.6	Connection paths between SARS-CoV-2 NSP7 (viral protein) and Protein:P62834 in the SARS-CoV-2 PPI knowledge graphs.	65
3.7	Bio-JOIE performance on different train-set ratios of SARS-CoV-2-Human PPIs.	65
3.8	A high-level overview of the script to the modularized workflow of ML applications.	68
3.9	Detailed steps of original knowledge graph triple completion problem. While all steps can be included in one single script, it can be decomposed into multiple modules that are easy to understand and control.	68

3.10	Inference steps on SARS-CoV-2 drug repurposing are successfully running on Texera. Each row represents one single query such as {Disease:SARS-CoV-2 ORF7a, Relation: Disease-Compound, Compound:?}. With one click, the entire information of the corresponding row will be presented in detail.	69
3.11	Biological knowledge graphs for drug-target discovery from BETA benchmark [ZLW22].	70
4.1	One application of complementary product recommendation in Amazon. Multiple complementary items are listed after one item has been put into the cart by the customer.	72
4.2	Three lists of “to-buy-together” recommendation on the e-commerce platform. Good complementary recommendations require both relatedness and diversity. .	72
4.3	One snapshot of a typical BPG. BPG is constructed with nodes as items with catalog features (type, etc) and edges as pairwise relations based on customer behavior.	77
4.4	Connections between the two-view KG and the two-view BPG.	78
4.5	Behavior based item relations have overlaps with each other. The overlaps of $\mathcal{B}_{cp} \cap \mathcal{B}_{cv}$ no doubt cast a challenge of complementary label correctness.	79
4.6	High complement transition ratio among co-purchase pairs in \mathcal{B}_{cp} in terms of categories and types.	79
4.7	P-Companion model architecture for complementary recommendation. As an embedding based recommender, it has three major components: type transition, type-item prediction, along with product2vec for pretrained product embeddings.	82
4.8	GNN-based Product2vec module architecture, which learns effective product embeddings given its textual features and aggregation from similar products. . . .	83
4.9	One MTurk survey snapshot for complements recommendation evaluation. MTurk workers are asked to tell their purchase willingness in a range of 0-3.	95

4.10	Comparison on Type Hit@3 and Item Hit@10 performance under different hyperparameter settings of α	100
5.1	Example of data to ontology matching.	106
5.2	MEDTO system architecture.	110
5.3	MEDTO ontology enrichment.	112
5.4	Details of MEDTO matching module. Both O_1 and O_2 are split into the hierarchical and non-hierarchical facets, which are fed into a hyperbolic graph layer and a heterogeneous graph layer respectively. The matching module minimizes the contrastive matching loss to let the representations of matching concepts have a very small distance while those of unmatched concepts have a large distance. . .	113
5.5	Local and global contexts of “renal failure”.	117
5.6	Sensitivity analysis.	125
6.1	Two examples of multi-source graph transfer in knowledge bases (left) and enterprise systems (right). By leveraging the entities and relations from sources $G_{S^{(1)}}$ and $G_{S^{(2)}}$, we can estimate the target graph \hat{G}_T based on the current observation G_T . Grey nodes/links in \hat{G}_T denote new predictions from graph knowledge transfer. (Best viewed in color)	129
6.2	Model architecture overview for MSGT-GNN (two source graphs are shown). Node embeddings across multiple graphs are learned through two-module framework, i.e. <i>Intra-graph encoder</i> , which learns node embeddings of its own graph context from initial node features; and <i>Cross-Graph transfer</i> , which enables learning through multiple graphs and node embeddings are updated by its corresponding nodes in other source as well as the graph-level information.	133
6.3	Details about Cross-Graph Transfer Layer operating on the Node T_i , updated by itself and its corresponding cross-graph neighbors (node-level embeddings), attentively learned from graph-level embeddings (Best viewed in color)	135

6.4	Performances with graph maturity. Most models achieve average F1 score close to 1 as the maturity of input observed target graph grows, while MSGT-GNN outperforms other baselines.	143
6.5	Performance comparison with different embedding dimensions d	144
6.6	Performance comparison with different values of μ , which explicitly balances the importance and reliability of source and target.	144
7.1	A snapshot of one chronological entry from a real-world homicide investigation case.	151
7.2	Example of the curated case-centric MKG schema with only node types presented.	152
7.3	Profile example of one (de-identified) vehicle evidence node with extracted descriptions.	154
7.4	M-BERT model workflow for private homicide-domain language models.	155
7.5	Architecture of hierarchical attention layers for case representation learning.	156
7.6	Number of chronological entries vs case duration in days since the incident for 40 randomly selected cases. Green/red lines denotes solved/unsolved cases respectively. (Best viewed in color)	158
7.7	Visualization of attention weights from multiple input entries on the classification of one single case. Given the first 5 entries as input, the entry with the observation of <i>Car</i> at the crime scene leads to the highest weight predicted by EIHA, instead of <i>Bike</i> . The last 3 chrono entries are provided for the full context of the case, but not as model input.	162
8.1	Inspired by progress in large-scale language modeling, DeepMind has proposed a multi-modal, multi-task, multi-embodiment generalist AI agent that can serve multiple inputs of images, text, and human-robot interactions and corresponding applications, ranging from gaming, chatbots, visual question answering.	171

LIST OF TABLES

1.1	Representative examples of knowledge graph embedding methods.	3
2.1	Statistics of datasets.	24
2.2	Results of KG triple completion. H@1 and H@10 denote <i>Hit@1</i> and <i>Hit@10</i> respectively. For each group of model variants with the same intra-view model, the best results are bold-faced. The overall best results on each dataset are underscored.	26
2.3	Results of entity typing on YAGO26K-906 and DB111K-174.	28
2.4	Examples of ontology population from JOIE-TransE-CT. Top 5 Populated Triples with smallest L2-norm distances are provided with reasonable answers bold-faced.	32
2.5	Results of long-tail entities typing.	34
2.6	Examples of long-tail entity typing. Top 3 predictions are provided with the correct type bold-faced.	34
2.7	Effects of negative sampling in type links	39
3.1	Statistics of PPI networks and associated GO annotations from different species.	54
3.2	Statistics of three aspects in the gene ontology: biological processes (BP), cellular components (CC) and molecular functions (MF).	54
3.3	PPI type prediction accuracy (%) evaluated on yeast, fly and human species.	57
3.4	Comparison of PPI prediction accuracy of Bio-JOIE on three different aspects of gene ontology.	59
3.5	PPI type prediction accuracy on different configurations of multi-species joint learning.	59
3.6	Results of top-level EC clustering by K-means on learning selected yeast protein embeddings.	61

3.7	F-1 score on SARS-CoV-2-Human PPI interaction classification.	63
3.8	Top target proteins predicted by Bio-JOIE. Known interactions from training set are excluded. Proteins that are considered as high-confidence targets are boldfaced.	64
4.1	Comparison between P-Companion and existing representative models: Co-Purchase, Sceptre and PMSC.	74
4.2	Comparison between KGs and BPGs.	78
4.3	Product item relationship analysis on combinations of \mathcal{B}_{cp} , \mathcal{B}_{cv} and \mathcal{B}_{pv} in terms of classification accuracy by human evaluation. on Electronics (Elec.), Grocery (Gro.) and all categories (All).	80
4.4	P-Companion notations used in Chapter 4	81
4.5	Dataset statistics.	89
4.6	Results of complementary recommendation based on historical FCP records on <i>Electronics</i> dataset.	91
4.7	Results of complementary recommendation based on historical FCP records on <i>Grocery</i> datasets.	92
4.8	Results of complementary recommendation based on historical FCP records on large-scale All-Group dataset. (H@k denotes Hit@k score.)	92
4.9	Performance of P-Companion with different number of predicted item types on Electronics and Grocery dataset.	92
4.10	MTurk comparison between P-Companion’s Top-5 recommendations and co-purchase record. Percentage of Score X represents the proportion of pairs labeled with Score-X.	96
4.11	Query item coverage comparison on all three datasets. P-Companion shows better product item coverage over co-purchase record.	96

4.12	Results on complementary recommendation on cold-start product items under electronics categories. (H@k denotes Hit@k score.)	99
4.13	Examples of complementary recommendation on cold-start items with P-Companion output. Recommendations are highly related and diverse even though the resource of co-purchase history is limited or unavailable.	99
4.14	Type transition examples. (Only top-3 transitions are listed for each type query.)	100
5.1	Matching MIMIC-III and MDX to SNOMED CT.	121
5.2	MDX-to-SNOMED result analysis (Hits@30).	121
5.3	MDX-to-SNOMED result analysis (Hits@30).	122
5.4	Results of ontology matching among FMA, NCI and SNOMED on OAEI datasets.	123
6.1	Summary of important notations.	132
6.2	Dataset statistics.	138
6.3	Results of KG triple completion. H@1 and H@10 denote <i>Hit@1</i> and <i>Hit@10</i> respectively. For each group of model variants with the same intra-view model, the best results are bold-faced. The overall best results on each dataset are underscored.	139
7.1	Results of case classification, comparing two categories of approaches. All shaded methods are developed in this work and the best results are bolded	161
7.2	Top-3 similar results retrieved by EIHA given one chrono entry in MurderBook. Tokens that are potentially highly related are shaded. Sensitive information have been anonymized or redacted.	168

ACKNOWLEDGMENTS

It has been an amazing journey to join UCLA's Computer Science PhD program and spend wonderful five years in the beautiful city of Los Angeles. Now the journey comes to an end, with countless stories, both up and down, joy and tears. The achievement of my PhD study, presented in this thesis could not be made possible without the help of so many people, my advisors, mentors and collaborators, my family and friends all along the way.

First of all, I would like to express my sincere gratitude to my PhD advisors, Prof. Wei Wang and Prof. Yizhou Sun, also as my doctoral committee co-chairs, for their continuous support, valuable advice, hands-on guidance, enthusiasm for teaching, and endless passion for research toward the frontier of machine learning and data mining, from Day 1. My marvelous journey all started with the opportunity they offered at UCLA as a new PhD student, which opened the door to a new world, though I had almost zero background in this area. In the past five years, they have always supported me with research ideas, fruitful collaborations across departments at UCLA, and career opportunities in the industry. They show me how to become an independent, responsible and self-motivated researcher all the time, from how to read papers, give constructive reviews, design and improve experiments, polish writing, to think of research and career plans, and pursue more innovations and breakthroughs.

In addition to my advisors, I would like to thank my committee members. Prof. Kai-Wei Chang and Prof. Ying Nian Wu for their discussions, feedback and suggestions. Their expertise and support always inspire me in my research throughout my PhD study. I also want to thank all my great instructors during my course study at UCLA from CS, ECE and Stat departments, especially Prof. Junghoo (John) Cho, Prof. Jonathan Kao and Prof. Lieven Vandenberghe. Other than major courses, I also enjoy my time spent learning Spanish, German, French and Arabic, with Ben Burt, Kevin Ziehl, Molly Courtney, Nick Smith, Unai Nafarrate and Chris Gobeille.

I am honored to work with many faculty members with Prof. Carlo Zaniolo, Prof. P. Jeffrey Brantingham (Department of Anthropology) and Prof. Chen Li (UCI Computer

Science), many outside of Computer Science department at UCLA. These collaborations provide me with valuable experiences and project management skills in interdisciplinary research from broader domains, which bring AI into real-world challenges. As a member of the ScAI lab at UCLA, I am grateful to work with many talented colleagues including Muhao Chen, Chelsea Ju, Jyun-Yu Jiang, Wenchao Yu, Xiushi Chen, Yunsheng Bai, Alexander R. Pelletier, and Alex Taylor. I enjoyed working with them and gained experience in conducting research and managing collaborations.

I am fortunate to have four internships during my PhD study. The internship projects result in many great and impactful works and contribute to several chapters in this thesis. I would like to express my greatest appreciation to all of my mentors and managers: Dr. Lu-An Tang (NEC Labs America), Dr. Tong Zhao, Dr. Jin Li and Dr. Luna Xin Dong (Amazon), Dr. Chuan Lei and Dr. Fatma Özcan (IBM Research Almaden) and Dr. Zhihong Shen, Chieh-Han Wu, Dr. Ye-Yi Wang and Prof. Jennifer Neville (Microsoft Research and MSAI). They expanded my research scope into the industrial environment and encouraged me to explore interesting projects. During my internships (two in-person and two virtual internships), I feel lucky to meet many distinguished researchers, outstanding engineers and other interns (also all as friends) and enjoy every minute of discussing with them and learning from their talks and presentations during “coffee chat”.

I always find myself passionate about teaching. Stepping into the classrooms (or Zoom) and giving two-hour lectures always excites me because conveying knowledge to younger “kids” is one of the most rewarding things. I am thankful for having the teaching opportunities for five quarters, working with David Smallberg, Prof. Yizhou Sun, and Prof. Sriram Sankararaman, some of the best CS instructors. Their well-designed classes encourage me to do better in delivering lectures and have taught me how to organize the “teaching” as research projects.

Also, I also want to say thanks to my undergraduate advisors at Department of Automation, Tsinghua University before joining UCLA, Prof. Xin Pei and Prof. Zuo Zhang. They brought me into the first step into the world called “research”, equipped me with solid

skillsets, and strongly encouraged me to continue my research further in the United States.

My PhD life is more colorful with the best memories with my friends I've met at UCLA: Wenchao Yu, Muhao Chen, Ting Chen, Yupeng Gu, Ruirui Li, Chelsea Ju, Yichao Zhou, Seungbae Kim, Cheng Zheng, Jyun-Yu Jiang, Zeyu Li, Jieyu Zhao, Guangyu Zhou, Xiusi Chen, Xuelu Shirley Chen, Nathan LaPierre, Yichao Zhou, Jin Wang, Yunsheng Bai, Jian Weng, Ruchen Zhen, Zhaoxing Deng, Kewei Cheng, Ziniu Hu, Song Jiang, Yewen Wang, Zhiping Xiao, Shichang Zhang, Zijie Huang, Dongruo Zhou, Roshni Iyer, Alex Pelletier, Alex Taylor, Yan Yu, Tao Meng, Liunian Harold Li, Da Yin, Changjun Fan, Chang Gao, Pei Han, Yongkai Liu, Danfeng Guo, Pei-Hung Chen, Weinan Song, Yaxuan Zhu, Xuanyi Wu, Yue Ariel Wu, Haoxin Zheng, Chenwei Gong, Manoj Reddy, Mingda Li, Zijun Xue, Shengming Zhang, Madelene Hem and many more. Also, I am very grateful to have more friends from my internship companies at NEC Labs America, Amazon, IBM, and Microsoft, as well as conferences and other opportunities: Bo Zong, Shasha Li, Chuxu Zhang, Jian Fang, Kai Li, Qi Wang, Hao Sun, Ning Xie, Shen Wang, Dongkuan Xu, Yao Li, Ziqiao Zhou, Chuhan Gao, Wajih Ul Hassan, Adil Ahmed, Qizhen Zhang, Hanbo Wang, Yan Liang, Yaqing Wang, Zhengyang Wang, Qian Zheng, Jiaqi Hao, Hao Wei, Giannis Karamanolakis, Jiawei Zhou, Sanxing Chen, Sonal Joshi, Zhen Wang, Zhuo Chen, Yining Wen, Qin Zhang, Hao Ding, Sijia Liu, Shouwei Hui, Hua Wei, Cheng Shen. Special thanks to my roommates over these years at LA, Tianxiang Li and Jyun-Yu Jiang, and "tennis pals" at UCLA courts (Tianxiang Li, Lingyu James Zhan, and Tong Lu) during the pandemic, even without a real net.

Last and most importantly, I want to say a huge thank you to my loving family. There is not enough word to express how much I owe to my father, Gang Hao, and my mother, Guoqiu Zhang, for their unconditional and endless love which gives me the strongest support through the darkest moments, crossing thousands of miles over the Pacific. PhD life is often like a lonely warrior. However, only when I see their smiles, I am still a kid, feeling warm and loved, not so far away from home.

A journey is about to end and another awaits. Always be grateful for what life gives, and always be ready for whatever it takes.

VITA

- 2013 – 2017 B. Eng. in Information Science and Technology, Tsinghua University
Beijing, China
- 2014 – 2017 B. Sc (Econ, Minor) in Economics and Management, Tsinghua University
Beijing, China
- 2018 Research Intern, NEC Laboratories America
Princeton, NJ, United States
- 2019 Applied Scientist Intern, Amazon.com Services
Seattle, WA, United States
- 2018 – 2021 Teaching Assistant/Associate, Computer Science Department, UCLA
Los Angeles, CA, United States
- 2020 Research Intern, IBM Research AI
San Jose, CA, United States
- 2021 Research Intern, Microsoft Research
Redmond, WA, United States
- 2017 – 2022 Graduate Student Researcher, Computer Science Department, UCLA
Los Angeles, CA, United States

PUBLICATIONS

Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, Wei Wang. “Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological

Concepts”, in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2019).

Junheng Hao, Chelsea J.-T. Ju, Muhao Chen, Yizhou Sun, Carlo Zaniolo, Wei Wang. “Bio-JOIE: Joint Representation Learning of Biological Knowledge Bases”, in Proceedings of the 11th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM-BCB 2020).

Junheng Hao, Tong Zhao, Jin Li, Luna Xin Dong, Christos Faloutsos, Yizhou Sun, Wei Wang. “P-Companion: Framework for Diversified Complementary Product Recommendation”, in Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020).

Jyun-Yu Jiang, Chelsea J.-T. Ju, **Junheng Hao**, Muhao Chen, Wei Wang. “JEDI: Circular RNA Prediction based on Junction Encoders and Deep Interaction among Splice Sites”, in Proceedings of the 29th annual international conference on Intelligent Systems for Molecular Biology and the 20th annual European Conference on Computational Biology (ISMB-ECCB 2021).

Junheng Hao, Chuan Lei, Abdul Quamar, Vasilis Efthymiou, Fatma Ozcan, Yizhou Sun, Wei Wang. “MEDTO: Medical Data to Ontology Matching using Hybrid Graph Neural Networks”, in Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2021).

Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, **Junheng Hao**, Ye-Yi Wang, Kuansan Wang and Jiawei Han. “Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification.” In Proceedings of The Web Conference (WWW 2022).

Junheng Hao, Lu-An Tang, Yizhou Sun, Zhengzhang Chen, Haifeng Chen, Junghwan Rhee, Zhichun Li and Wei Wang. “Multi-source Inductive Graph Knowledge Transfer”, in Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD 2022).

CHAPTER 1

Introduction

1.1 Knowledge Bases and Graphs

Knowledge Bases (KBs) and Knowledge Graphs (KGs)¹, such as Wikipedia, have large numbers of real-world facts and have rapidly become a mainstream technology that combines features of databases and AI and are foundational to many knowledge-driven AI applications. Considering one of the most representative KGs, Wikipedia, is a semi-structured KG with billions of entities and thousands of relations between entities, covering all aspects of the world, from tennis tournaments to political figures, from historical places to airports. Such knowledge can be explicitly constructed by triples as its atomic component formatted as, such as {Washington D.C., capital city of, the United States}, which is typically named as RDFs [AH11]. Alternatively, it is more often implicitly stored and represented as semi-structured “InfoBox” [WW08], sometimes referred to as “Labeled-Property Graph” with multiple attributes (textual descriptions, along with entity relations. Other examples are general-purpose KGs (Wikidata [VK14], YAGO [MBS14, SKW07, PWS20, HSB13], DBpedia [LIJ15], Freebase [BEP08]), commonsense NLP-related KGs (WordNet [Mil95], ConceptNet [SCH17]), domain-specific KGs (STRING [SMC16]) and enterprise KGs (examples Amazon Catalog Product Graphs [MPL15], behavior-based Product Graphs [HZL20]) and many more.

¹Knowledge graphs and knowledge bases may have different definitions and references among different research communities. Typically, knowledge graphs emphasize the “graph” nature with entities (nodes), relations (edges), and logical triples to represent real-world facts; sometimes knowledge bases can be referred to as (graph) databases and systems, which may not be in the format of graphs. In this paper, we use knowledge graphs and knowledge bases interchangeably and in most cases towards the “knowledge graphs” unless specified otherwise.

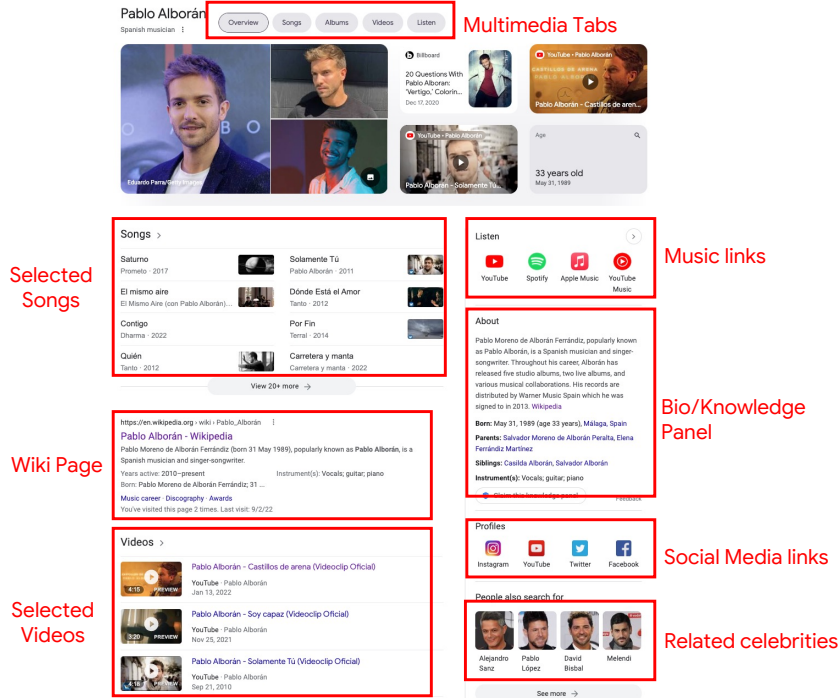


Figure 1.1: Searching results of “Pablo Alborán” returned by Google, as of Aug 31, 2022.

These KBs provide a large amount of high-quality prior knowledge, describe relational facts or interactions among entities form an important role in knowledge acquisition and serve as the foundation for many knowledge-driven AI tasks and applications. As one of the most important applications, knowledge graphs are no doubt one of the foundations in web-scale search engines which collect and arrange all types of facts of one specific entity to provide accurate and timely searching results. For example, when one user search for “Pablo Alborán” (a famous Spanish singer) in Google, you will get the following page in Figure 1.1 which is a comprehensive collection of his public information across multiple websites together with social media, sometimes together with multimedia resources (music, video, images, etc) and new headlines, even with potential matched advertisements. The knowledge backend, even not in the “graph” format, is essential to make such profiles and provides highly-related information that aligns with the user’s interest.

1.2 Machine Learning on Knowledge Graphs

Machine learning techniques have been continuously in development on knowledge graph, which is an intersection research area among graph machine learning, database and data management, ontology and semantic web, and natural language processing. With the rise of deep learning, knowledge graph embedding has been since [BUG13]. In the past decade, KG embedding models have been widely investigated. As deep learning based techniques on knowledge graphs, these embedding models, which typically encode KG structures into low-dimensional embedding spaces, are vital to capturing the latent semantic relations of entities and concepts and support relational inferences in the form of vector algebra. More specifically, the aim is to design a score function representing the plausibility of one relational fact (i.e. triple), including translation-based, similarity-based, or even more complex models with CNN and Transformers. Some representative examples of methods in this thread are listed in Table 1.1.

Table 1.1: Representative examples of knowledge graph embedding methods.

Model	Score Function	Embeddings
TransE (Bordes et al., 2013)	$- \mathbf{h} + \mathbf{r} - \mathbf{t} $	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
TransX	$- g_{r,1}(\mathbf{h}) + \mathbf{r} - g_{r,2}(\mathbf{t}) $	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
DistMult (Yang et al., 2014)	$(\mathbf{h} \circ \mathbf{t}) \cdot \mathbf{r}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
HolE (Nickel et al., 2016)	$(\mathbf{h} \star \mathbf{t}) \cdot \mathbf{r}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
ComplEx (Trouillon et al., 2016)	$\text{Re}\langle \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k$
ConvE (Dettmers et al., 2017)	$\langle \sigma(\text{vec}(\sigma([\mathbf{r}, \mathbf{h}] * \Omega))\mathbf{W}), \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$
RotatE (Sun et al., 2019)	$- \mathbf{h} \circ \mathbf{r} - \mathbf{t} ^2$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^k, r_i = 1$

In addition, learning knowledge graphs is closely related to heterogeneous information networks (HIN), such as citation networks with multiple types of nodes (authors, venues and institutions, etc) and their connections. Many techniques are also transferred and leveraged such as meta-path [SHY11, SH12] based methods to perform similarity search, relation prediction and alternative tasks.

Neural networks have been adapted to leverage the structure and properties of graphs. Another important advance in knowledge graph modeling is associated with graph neural

networks, which originated from network embedding and social network analysis. Originally from graph convolutional network on homogeneous graphs, many new GNN models are proposed on heterogeneous graphs [HDW20a, HDW20b] and knowledge graphs [SKB18, VSN19, LCC19] and also hierarchically structured “taxonomy”-like graphs [NK17, CYR19].

With the recent rapid rise of Transformer [VSP17], which has revolutionized natural language processing together with other domains and enabled multi-modality learning with a combination of vision and language, knowledge graphs as one unique format of modality have also been incorporated into contextual language models [SK21]. This results in many recent technical breakthrough on KG-augment language models [YTY21, LWH21, LZZ20, YHZ22], and knowledge probing [PRR19] and extraction from unstructured text [WDR21, AOO20]. State-of-the-art methods also include an integration of transformers and graph neural networks [ZBY22].

1.3 Knowledge Graph Empowered Applications

As mentioned in Section 1.1 and 1.2, knowledge graphs can help significantly improve the performance of many downstream applications [WMW17], including its connection to natural language processing [SK21] in many its sub-fields, especially in knowledge-intensive tasks [YDC22]. Examples are question answering [YRB21, ZBY22, MCL22], document understanding [ZSW22], entity recognition [AOO20, NGP21] and alignment [SZH20], ontology engineering and knowledge construction [HLE21], natural language understanding [LZZ20, GLT20], generation [YZL22, KJR21, YZQ22] and commonsense reasoning [JKH20, LWH21], and even into broader applications, recommender systems [HZL20, WSZ20], bioinformatics and healthcare [JJH21, HJC20, NYH16], traffic analysis and control [DSW21, SPH18, SHP16], and societal crime studies [PBU20, AGK20, LNM17]. It serves the role of “brain” in many knowledge-intensive AI applications to provide inferences based on real-world facts. Figure 1.2 shows some applications and directions backed by knowledge graphs, some covered in detail in later chapters of this dissertation.

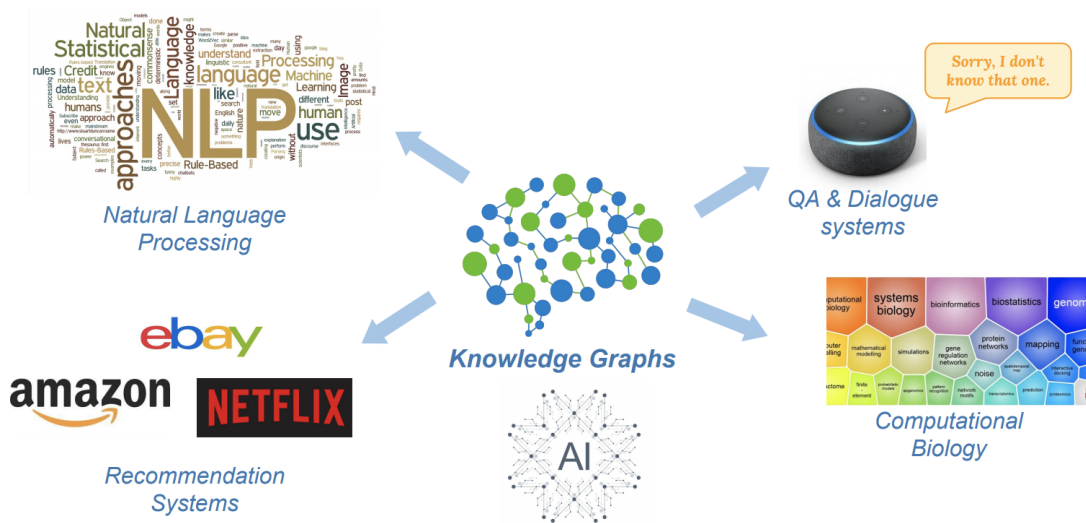


Figure 1.2: Examples of multidisciplinary applications with KG backends.

1.4 Research Roadmap and Thesis Contribution

In this dissertation, we emphasize the importance and benefit of modeling internal ontological structures inside knowledge graphs (such as type association and hierarchical ontology) into KG learning models. Accordingly, we successfully propose a series of machine learning methods that further investigate the capability of modeling such semantics in the KGs and demonstrate their effectiveness in multiple interdisciplinary applications.

As mentioned above, our contributions are in two threads: the technique thread and the application thread. We highlight the research development roadmap in this dissertation in Figure 1.3. We have explored *technique*-thread with innovative methods including, knowledge graph embedding [WMW17, HCY19], graph neural networks (including graph attention networks, relation GCN, and hyperbolic GCN), pretrained language models [QDM18, LYF21], self-supervised learning [JBZ20], multi-task learning, transfer learning, etc. As a result, these new techniques empowered a wide range of applications in knowledge graph inference, recommender systems, ontology matching, protein interaction and drug discovery (bioinformatics), document intelligence, and crime studies on homicide analytics, as in our *application*-thread. A detailed walkthrough of each project in the chapters of this dissertation is in Section 1.5.

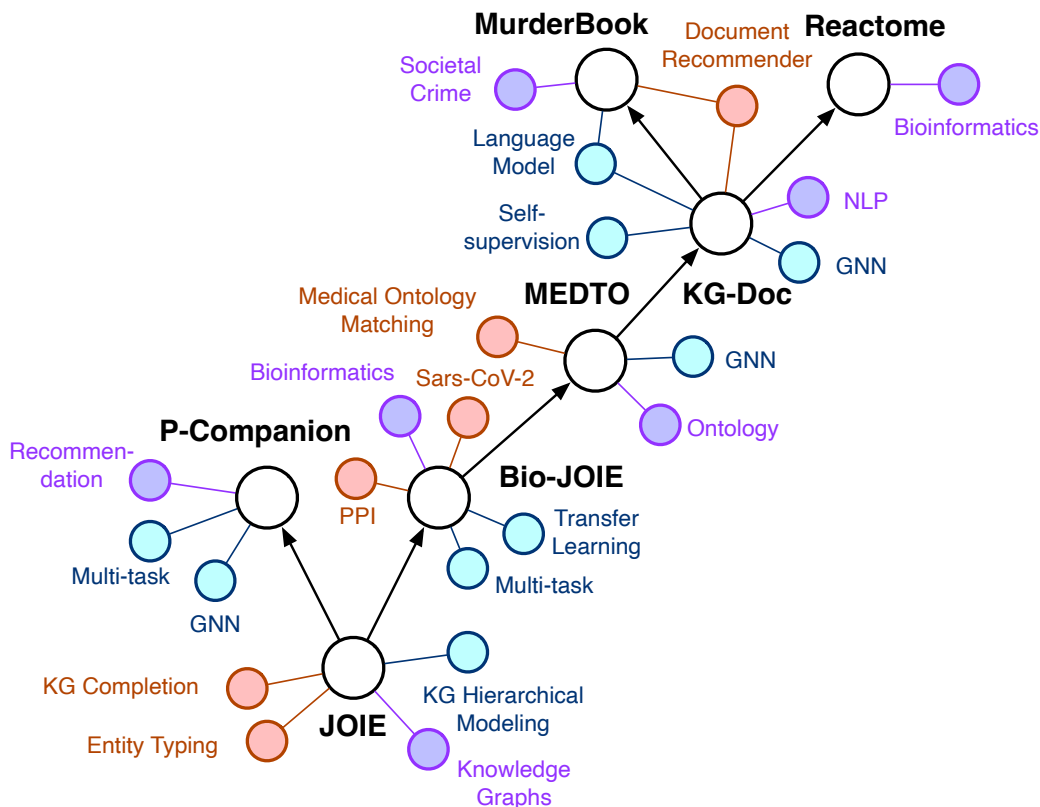


Figure 1.3: Research roadmap of this dissertation in the format of one knowledge graph, starting from bottom left JOIE node. White, red, blue and purple nodes represent main projects covered in this dissertation, applications/tasks, application domains, and proposed techniques/technical components respectively.

1.5 Thesis Overview

The rest of this dissertation can be organized into the following parts, in accordance with the research roadmap (Figure 1.3).

Chapter 2: We introduce JOIE, as in “*Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts*” [HCY19]. JOIE employs both cross-view and intra-view models that learn on multiple facets of the knowledge base. The cross-view association model is learned to bridge between the embeddings of ontological concepts and their corresponding instance-view entities. The intra-view models are trained to capture the structured knowledge of instance and ontology views in separate embedding spaces, with a hierarchy-aware encoding technique, enabled for ontologies with latent hierar-

chies. Our model is trained on large-scale knowledge bases that consist of massive instances and their corresponding ontological concepts connected via a (small) set of cross-view links. Experimental results on public datasets show that the best variant of JOIE significantly outperforms previous models on instance-view triple prediction task as well as ontology population on ontology-view KG. In addition, our model successfully extends the use of KG embeddings to entity typing with promising performance.

Chapter 3: As one important extension of JOIE in bioinformatics, we develop Bio-JOIE, as in “Joint Representation Learning of Biological Knowledge Bases” [HJC20]. Similar to the previous work, we propose the transferred multi-relational embedding model Bio-JOIE to capture the knowledge of gene ontology and PPI networks, which demonstrates superb capability in modeling the SARS-CoV-2-human protein interactions. Bio-JOIE jointly trains two model components. The *knowledge model* encodes the relational facts from the protein and GO domains into separated embedding spaces, using a hierarchy-aware encoding technique employed for the GO terms. On top of that, the *transfer model* learns a non-linear transformation to transfer the knowledge of PPIs and gene ontology annotations across their embedding spaces. By leveraging only structured knowledge, Bio-JOIE significantly outperforms existing state-of-the-art methods in PPI type prediction on multiple species. Furthermore, we also demonstrate the potential of leveraging the learned representations on clustering proteins with enzymatic function into enzyme commission families. Finally, we show that Bio-JOIE can accurately identify PPIs between the SARS-CoV-2 proteins and human proteins, providing valuable insights for advancing research on this new disease.

Chapter 4: As another extension of JOIE in the recommender systems, we develop “A Principled Framework for Diversified Complementary Product Recommendation” [HZL20] based on customer behavior based knowledge graphs, P-Companion, to explicitly model both relevance and diversity. More specifically, given one product with its product type, P-Companion first uses an encoder-decoder network to predict multiple complementary product types, and then a transfer metric learning network is developed to project the embedding of query product to each predicted complementary product type subspace and further learn

the complementary relationship based on the distant supervision labels. The whole framework can be trained from end-to-end and is robust to cold-start products attributed to a novel pretrained product embedding module named `Product2vec`, based on graph attention networks. Extensive offline experiments show that `P-Companion` outperforms state-of-the-art baselines by a 7.1% increase on the Hit@10 score with well-controlled diversity. Production-wise, we deploy `P-Companion` to provide online recommendations for over 200M products at Amazon and observe significant gains in product sales and profit.

Chapter 5: We formulate an innovative data-to-ontology problem, as a slight shift from general purpose knowledge graph to medical ontologies, and provide our solution `MEDTO` as in “*MEDTO: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks*”. Data to ontology matching is the process of finding semantic correspondences between tables in databases to standard ontologies. We design a novel end-to-end framework that consists of three innovative techniques: (1) a lightweight yet effective method that bootstrap a semantically rich ontology from a given medical database, (2) a hyperbolic graph convolution layer that encodes hierarchical concepts in the hyperbolic space, and (3) a heterogeneous graph layer that encodes both local and global context information of a concept. Experiments on two real-world medical datasets matching against SNOMED CT show significant improvements compared to the state-of-the-art methods. `MEDTO` also consistently achieves competitive results on a benchmark from the Ontology Alignment Evaluation Initiative.

Chapter 6: Large-scale information systems, as one type of knowledge graphs, exhibit dynamic and complex activities. Formalizing these information systems as graphs can effectively characterize the entities (nodes) and their relationships (edges). Transferring knowledge from existing well-curated source graphs can help construct the target graph of newly-deployed systems faster and better which no doubt will benefit downstream tasks such as link prediction and anomaly detection for new systems. we propose `MSGT-GNN` in this chapter “*Multi-source Knowledge Graph Transfer*”, a graph knowledge transfer model for efficient graph link prediction from multiple source graphs. `MSGT-GNN` consists of two components: the *Intra-Graph Encoder*, which embeds latent graph features of system entities into vectors;

and the graph transferor, which utilizes graph attention mechanism to learn and optimize the embeddings of corresponding entities from multiple source graphs, in both node level and graph level. Experimental results on multiple real-world datasets from various domains show that **MSGT-GNN** outperforms other baseline approaches in the link prediction and demonstrate the merit of attentive graph knowledge transfer and the effectiveness of **MSGT-GNN**.

Chapter 7: Homicide investigations produce a large amount of data, mostly text-intensive structural data including interviews, reports, descriptions of evidence, and summary statements. Performing insightful analytics based on such complex data plays a pivotal role in solving difficult homicide cases. In this chapter “*Empowering Homicide Analytics with MurderBook Knowledge Graphs and Domain-specific Language Models*”, we proposed a deep learning based systematic framework, **EIHA**, for multiple downstream applications such as case classification. The contributions are three-fold: first, we show how to extract various entity types and construct knowledge graphs that preserve rich text features alongside structured relational knowledge facts about the case (**KG** module); second, we introduce domain language models, developed on large-scale crime investigation summaries, which better model crime-related textual data (**LM** module); third, we learn comprehensive crime case representations, by utilizing a novel hierarchical attention mechanism, supported by the pillars of **KG** and **LM** modules. Experimental results show the effectiveness of case embeddings learned from **EIHA** in performing case classification. We also demonstrate two valuable application scenarios empowered by **EIHA** for AI-assisted analytics and data mining on homicide investigations.

Chapter 8: As the final part of this dissertation, we conclude our contributions with a summary of our research works.

CHAPTER 2

Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts

2.1 Introduction

As mentioned in Chapter 1, knowledge bases (KBs), like DBpedia [LJJ15], YAGO [MBS14] and ConceptNet [SCH17], have incorporated large-scale multi-relational data and motivated many knowledge-driven applications. These KBs store knowledge graphs (KGs) 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, KBs 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 KB.

Learning to represent a KB 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 an instance is rarely observed. For example, for a musician who has few relational facts in the instance-view graph, we can still tell his or her rough position in

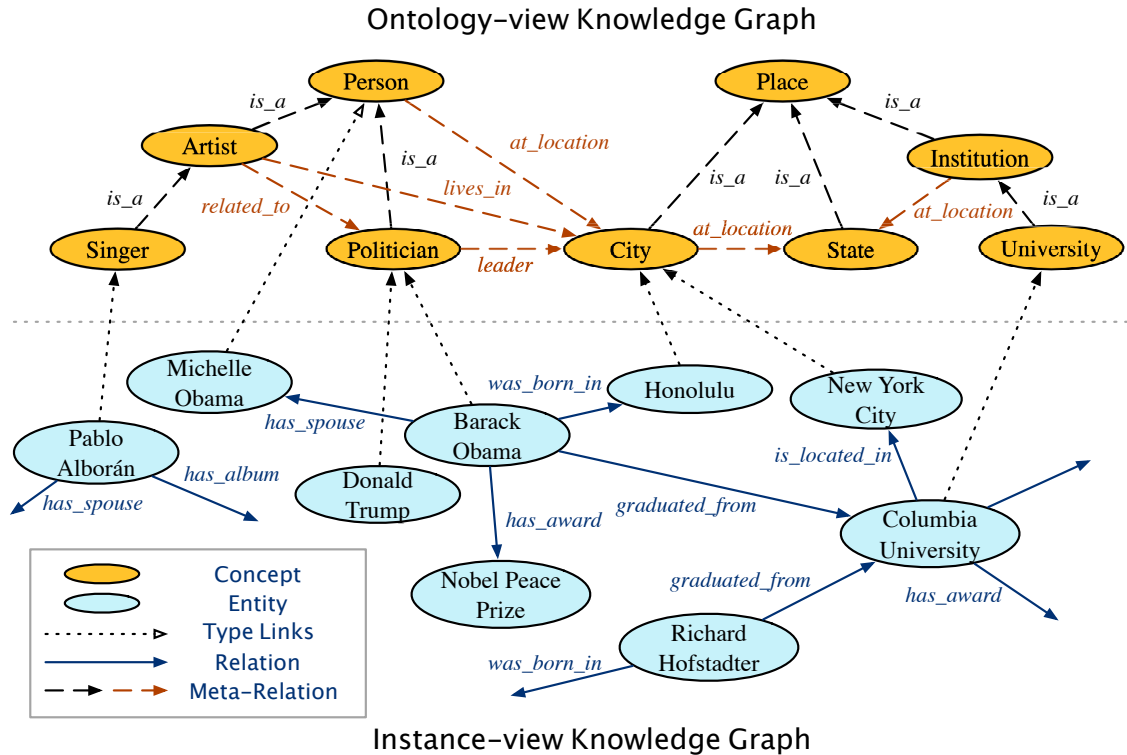


Figure 2.1: An example of two-view KB. Regular meta-relations and hierarchical meta-relations are denoted as orange and black dashed lines respectively in the ontology view.

instance embedding space because he or she should not be far away from the embeddings of other musicians.

In this chapter, we propose to jointly embed the instance-view graph and the ontology-view graph, by leveraging (1) triples in both graphs and (2) cross-view links that connect the two graphs. It is a non-trivial task to effectively combine representation learning techniques on both views of a KB together, which faces the following challenges: (1) the vocabularies of entities and concepts, as well as relations and meta-relations, are disjoint but semantically related in these two views of the KB, and 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; (2) the known cross-view links often inadequately cover a vast number of entities, which leads to insufficient information to align both views of the KB, and curtails discovering new cross-view links; (3) the scales and topological structures are also largely different in the two views, where the ontological views are often sparser,

provide fewer types of relations, and form hierarchical substructures, and the instance view is much larger and with much more relation types.

To address the above issues, we propose a novel KG embedding model named JOIE, which jointly encodes both the ontology and instance views of a KB. JOIE contains two components. First, a *cross-view association model* is designed to associate the instance embedding to its corresponding concept embedding. Second, the *intra-view embedding model* characterizes the relational facts of ontology and instance views in two separate embedding spaces. For the cross-view association model, we explore two techniques to capture the cross-view 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 intra-view 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 KBs where ontologies contain hierarchical substructures, we employ a *hierarchy-aware* embedding technique based on intra-view non-linear transformations to preserve such substructures. Accordingly, we investigate 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 KGs, which has significantly outperformed various baseline models. The results on the entity typing task show that our model is competent in discovering cross-view links to align the ontology-view and the instance-view KGs.

The rest of the chapter is organized as follows. We first discuss the related work in Section 2.2, then introduce the proposed JOIE model in Section 2.3. Section 2.4 presents the experiment evaluation together with case study and ablation study and the we concludes the chapter in Section 2.5.

2.2 Related Work

To the best of our knowledge, there is no previous work on learning to embed two-view knowledge of a KB. We discuss the following three lines of research work that are closely relevant to this work.

Knowledge Graph Embeddings. Recent work has put extensive efforts in learning instance-view KG embeddings. Given triples (h, r, t) , where r represents the relation between the head entity h and the tail entity t , the key of KG embeddings is to design a plausibility scoring function $f_r(\mathbf{h}, \mathbf{t})$ as the optimization objective (\mathbf{h} and \mathbf{t} are embeddings of h and t). A recent survey [WMW17] categorizes the majority of KG embedding models into translational models and similarity-based models. The representative translational model, TransE [BUG13], adopts the score function $f_r(\mathbf{h}, \mathbf{t}) = -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ to capture the relation as a translation vector \mathbf{r} between two entity vectors. Follow-ups of TransE typically vary the translation processes in different forms of relation-specific spaces, so as to improve the performance of triple completion. Examples include TransH [WZF14], TransR [LLS15], TransD [JHX15] and TransA [JWL16], etc. As for the similarity-based models, DistMult [YYH15] associates related entities using Hadamard product of embeddings, and HolE [NRP16] substitutes Hadamard product with circular correlation to improve the encoding of asymmetric relations, and achieves the state-of-the-art performance in KG completion. ComplEx [TWR16] migrates DistMult in a complex space and offers comparable performance. Besides, there are other forms of models, including tensor-factorization-based RESCAL [NTK11], and neural models NTN [SCM13] and ConvE [DMS18]. These approaches also achieve comparable performances on triple completion tasks at the cost of high model complexity.

It is noteworthy that a few approaches have been proposed to incorporate complex type information of entities into above KG embedding techniques [KBT15, XLS16, MDJ17, MCW18], from which our settings are substantially different in two perspectives: (i) These studies utilize the proximity of entity types to strengthen the learning of instance-level entity similarity, while do not capture the semantic relations between such types; (ii) They mostly

focus on improving instance-view triple completion, but do not leverage instance-view knowledge to improve ontology population, nor support cross-view association to bridge instances and ontological concepts. Another related branch on leveraging logic rules [RSR15, GWW16, DQW17] requires additional information that typically is not provided in two-view KBs.

Multi-graph Embeddings for KGs. More recent studies have extended embedding models to bridge multiple KG structures, typically for multilingual learning. MTransE [CTY17] thereof, jointly learns a transformation across two separate translational embedding spaces, which can be adopted to our problem. However, since this multilingual learning approach partly relies on similar structures of KGs, it unsurprisingly falls short of capturing the associations between the two views of KB with disjoint vocabularies and different topologies, as we show in the experiments. Later extensions of this model family, such as KDCoE [CTC18a] and JAPE [SHL17], require additional information of literal descriptions [CTC18a] and numerical attributes of entities [SHL17] that are typically not available in the ontology views of the KB. Other models depend on the use of neural machine translation [OKK18], causal reasoning [YWC18] and bootstrapping of strictly 1-to-1 matching of inter-graph entities [ZXL17, SHZ18] that do not apply to the nature of our corpora and tasks.

Ontology Population. Traditional ontology population is mostly based on extensive manual efforts, or requires large annotated text corpora for the mining of the meta-relation facts [WLB06, CS04, MAG14, GG08]. These previous approaches rely on intractable parsing or human efforts, which generate massive relation facts that are subject to frequent conflicts [PR13]. A few studies extend embedding techniques to general cross-domain ontologies like ConceptNet. Examples of such include On2Vec [CTC18b] that extends translational embeddings to capture the relational properties and hierarchies of ontological relations, and [GS18] propose to learn second-order proximity of concepts by combining chained logic rules with ontology embeddings. This shows the benefits of KG embeddings on predicting relational facts for ontology population, while we argue that such a task can be simultaneously enhanced with the characterization of the instance knowledge.

2.3 Modeling

In this section, we introduce our proposed model JOIE, which jointly embed entities and concepts using two model components: *cross-view association model* and *intra-view model*. We start with the formalization of two-view knowledge bases.

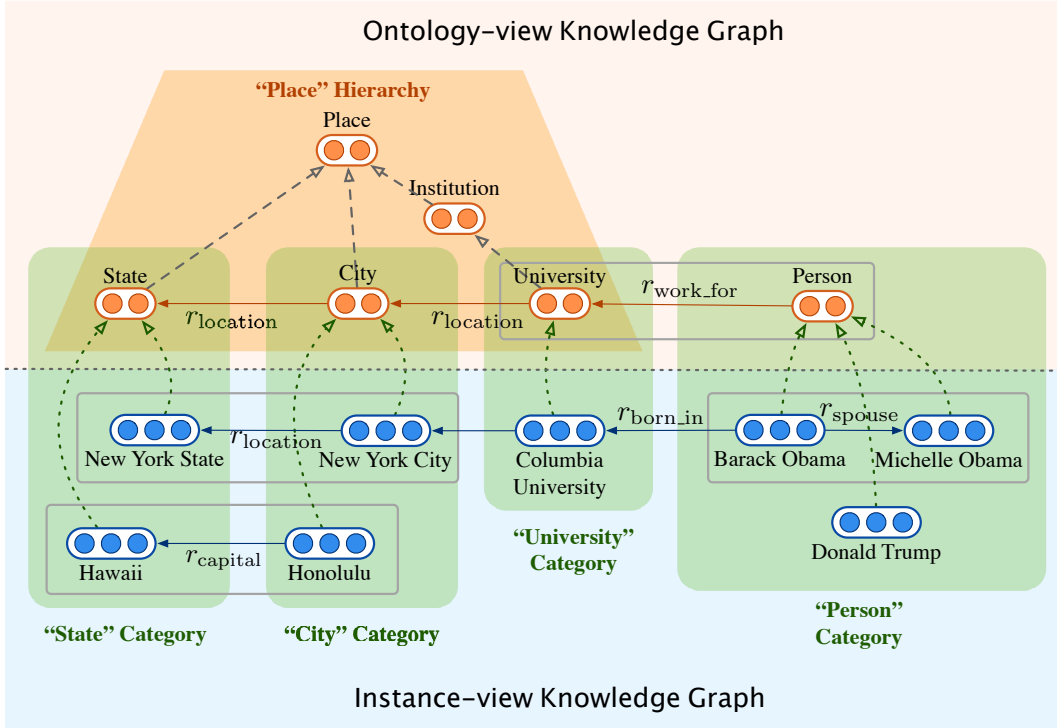


Figure 2.2: JOIE learns two aspects of a KB. The cross-view association model learns embeddings from cross-view links (dash arrows in green “category” box). The default intra-view model learns embeddings from triples (grey box) in each view; Besides, hierarchy-aware intra-view models the meta-relation facts that form hierarchies in the ontology (orange “Hierarchy” trapezoid).

2.3.1 Formalization of Knowledge Bases

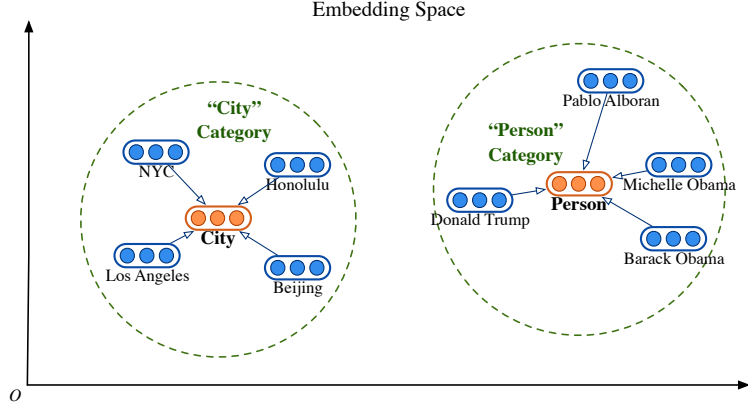
In a KB, we use \mathcal{G}_I and \mathcal{G}_O to denote the instance-view KG and ontology-view KG respectively. The instance-view KG is denoted as \mathcal{G}_I , which is formed with \mathcal{E} , the set of entities, and \mathcal{R}_I , the set of relations. The set of concepts and meta-relations 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

KG and the ontology-view KG 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 KB, 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 instance-view triples. Similarly, $\mathbf{h}^{(O)}, \mathbf{t}^{(O)}$, and $\mathbf{r}^{(O)}$ denote the embedding vectors for the corresponding concepts and their meta-relation in the ontology-view graph. Besides the notations for two views, \mathcal{S} is used to denote the set of known cross-view links in the KB, 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, $(e: \text{Los Angeles International Airport}, c: \text{airport})$ 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 other similar meta-relations). That is, we can observe concept pairs $(c_l, c_h) \in \mathcal{T}$ that indicates a finer (more specific) concept belongs to a coarser (more general) concept. One aforementioned example is $(c_l: \text{singer}, c_h: \text{person})$.

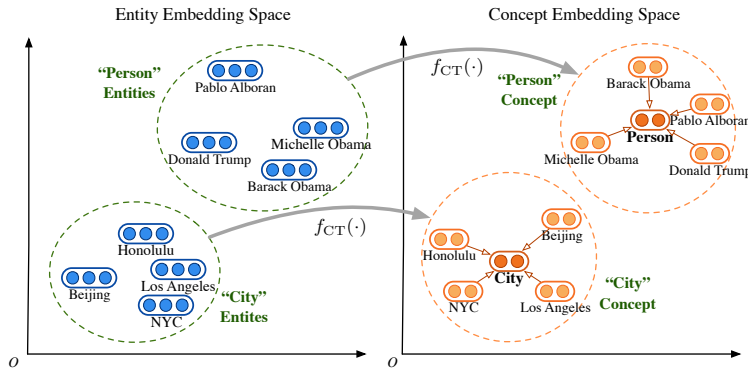
Our model JOIE consists of two model components that learn embeddings from the two views: the cross-view association model enables the connection and information flow between the two views by capturing the instantiation of entities from corresponding concepts, and the intra-view model encodes the entities/concepts and relations/meta-relations on each view of the KB. The illustration of these model components for learning different aspects of the KB is shown in Figure 2.2. In the following subsections, we first discuss the cross-view association model and intra-view model for each view, then combine them into variants of proposed JOIE model.

2.3.2 Cross-view Association Model

The goal of the cross-view association model is to capture the associations between the entity embedding space and the concept embedding space, based on the cross-view links in KBs, 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



(a) Cross-view Grouping (CG)



(b) Cross-view Transformation (CT)

Figure 2.3: Intuition of the cross-view association model: Cross-view Grouping (a); Cross-view Transformation (b).

based on different assumptions and thus optimize different objective functions.

Cross-view Grouping (CG). The cross-view grouping method can be considered as grouping-based regularization, which assumes that the ontology-view KG and instance-view KG 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 2.3a. 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,

$$J_{\text{Cross}}^{\text{CG}} = \frac{1}{|\mathcal{S}|} \sum_{(e,c) \in \mathcal{S}} [\|c - e\|_2 - \gamma^{\text{CG}}]_+, \quad (2.1)$$

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} -radius¹ 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 2.3b. 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:

$$\mathbf{c} \leftarrow f_{\text{CT}}(\mathbf{e}), \forall (e, c) \in \mathcal{S}, \quad (2.2)$$

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}_{ct} is a bias vector. $\sigma(\cdot)$ is a non-linear activation function, for which we adopt tanh. Therefore, the total loss of the cross-view association model is formulated as Equation 2.3, which aggregates the CT objectives for all concepts involved in \mathcal{S} .

$$J_{\text{Cross}}^{\text{CT}} = \frac{1}{|\mathcal{S}|} \sum_{\substack{(e,c) \in \mathcal{S} \\ \wedge (e,c') \notin \mathcal{S}}} [\gamma^{\text{CT}} + \|\mathbf{c} - f_{\text{CT}}(\mathbf{e})\|_2 - \|\mathbf{c}' - f_{\text{CT}}(\mathbf{e})\|_2]_+ \quad (2.3)$$

2.3.3 Intra-view Model

The aim of intra-view model is to preserve the original structural information in each view of the KB 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,

¹Typically, 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.

improving the performance of downstream tasks, as shown in Section 2.4.2. In this section, we provide two intra-view model techniques for encoding heterogeneous and hierarchical graph structures.

Default Intra-view Model. To embed such a triple (h, r, t) in one KG, 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 intra-view framework. We adopt three representative techniques, i.e. translations [BUG13], multiplications [YYH15] and circular correlation [NRP16]. The score functions of these techniques are given as follows.

$$\begin{aligned} f_{\text{TransE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= -\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_2 \\ f_{\text{Mult}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= (\mathbf{h} \circ \mathbf{t}) \cdot \mathbf{r} \\ f_{\text{HolE}}(\mathbf{h}, \mathbf{r}, \mathbf{t}) &= (\mathbf{h} \star \mathbf{t}) \cdot \mathbf{r} \end{aligned} \tag{2.4}$$

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

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

$$J_{\text{Intra}}^{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{\substack{(h,r,t) \in \mathcal{G} \\ \wedge (h',r,t') \notin \mathcal{G}}} [\gamma^{\mathcal{G}} + f(\mathbf{h}', \mathbf{r}, \mathbf{t}') - f(\mathbf{h}, \mathbf{r}, \mathbf{t})]_+, \tag{2.5}$$

where $\gamma^{\mathcal{G}} > 0$ 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 intra-view 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 $J_{\text{Intra}}^{\mathcal{G}_I}$ and $J_{\text{Intra}}^{\mathcal{G}_O}$. Combining the loss from instance-view and ontology-view graphs, the joint loss of the intra-view model is given as below,

$$J_{\text{Intra}} = J_{\text{Intra}}^{\mathcal{G}_I} + \alpha_1 \cdot J_{\text{Intra}}^{\mathcal{G}_O}, \tag{2.6}$$

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 Intra-view model, we employ the same triple encoding technique to represent both views of the KB. 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 KG embeddings, which can potentially be used in our intra-view model. Since exploring different triple encoding techniques is not the focus of our motivation, we leave them as future work.

Hierarchy-Aware Intra-view Model for the Ontology. It is observed that the ontology view of some KBs form hierarchies, which is typically constituted by a meta-relation with the hierarchical property, such as “*subclass_of*” and “*is_a*” [MBS14, LIJ15]. We can define such meta-relation facts as $(c_l, r_{\text{meta}} = \text{“subclass_of”}, c_h)$. 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 intra-view model.

To address this problem, we propose the hierarchy-aware (HA) intra-view model by extending a similar method to that of cross-view transformation as defined in Equation 2.2. Given concept pairs (c_l, c_h) , we model such hierarchies into a non-linear transformation between coarser concepts and associated finer concepts by

$$g_{\text{HA}}(\mathbf{c}_h) = \sigma(\mathbf{W}_{\text{HA}} \cdot \mathbf{c}_l + \mathbf{b}_{\text{HA}}) \quad (2.7)$$

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 2.3,

$$J_{\text{Intra}}^{\text{HA}} = \frac{1}{|\mathcal{T}|} \sum_{\substack{(c_l, c_h) \in \mathcal{T} \\ \wedge (c_l, c_h') \notin \mathcal{T}}} [\gamma^{\text{HA}} + \|\mathbf{c}_h - g(\mathbf{c}_l)\|_2 - \|\mathbf{c}_h' - g(\mathbf{c}_l)\|_2]_+ \quad (2.8)$$

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

$$J_{\text{Intra}} = J_{\text{Intra}}^{\mathcal{G}_I} + \alpha_1 \cdot J_{\text{Intra}}^{\mathcal{G}_O \setminus \mathcal{T}} + \alpha_2 \cdot J_{\text{Intra}}^{\text{HA}} \quad (2.9)$$

where positive α_1 and α_2 are two weighing hyperparameters. In Equation 2.9, $J_{\text{Intra}}^{\mathcal{G}_O \setminus \mathcal{T}}$ refers to the loss of the default intra-view model that is only trained on triples with regular semantic relations. $J_{\text{Intra}}^{\text{HA}}$ is explicitly trained on the triples with meta-relations that form the ontology hierarchy, which is a major difference from Equation 2.6.

As the conclusion of this subsection, in JOIE, the basic assumption is that KGs have ontology hierarchy and rich semantic relational features compared to social or citation networks. JOIE is able to encode such KG 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.

2.3.4 Joint Training on Two-View KBs

Combining the intra-view model and cross-view association model, JOIE minimizes the following joint loss function:

$$J = J_{\text{Intra}} + \omega \cdot J_{\text{Cross}}, \quad (2.10)$$

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

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

We use the AMSGrad optimizer [RKK18] 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 [SMG14]. 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 [BUG13, YYH15, WZF14, NRP16]. Negative sampling is used on both intra-view model and cross-view association model with a ratio of 1 (number of negative samples per positive one). A hinge loss is applied for both models with all variants.

2.3.5 Variants of JOIE and Complexity

Without considering the HA technique, we have six variants of JOIE given two options of cross-view association models in Section 2.3.2 and three options of intra-view models in Section 2.3.3. 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 intra-view embeddings. In addition, we incorporate the hierarchy-aware intra-view model for the ontology view into cross-view transformation model², which produces three additional model variants denoted as JOIE-HATransE-CT, JOIE-HAMult-CT, and JOIE-HAHolE-CT.

The model complexity depends on the cross-view association model and intra-view model for learning two-view KBs. 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 intra-view 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 KG. For the task of triple completion in the KG, 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 $\mathcal{O}(n_e d_e)$ for CG and $\mathcal{O}(n_c d_c d_e)$ for CT. Details about each task are curated in Section 2.4.2 and 2.4.3.

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

2.4 Experiments

In this section, we evaluate JOIE with two groups of tasks: the triple completion task (Section 2.4.2) on both instance-view and ontology-view KGs and the entity typing task (Section 2.4.3) to bridge two views of the KB. Besides, we provide a case study in Section 2.4.4 on ontology population and long-tail entity typing. We also present hyperparameter study, effects of cross-view sufficiency and negative samples in Section 2.4.5.

2.4.1 Datasets

To the best of our knowledge, existing datasets for KG embeddings consider only an instance view (e.g. FB15k [BUG13]) or an ontology view (e.g. WN18 [BGW14]). Hence, we prepare two new datasets: *YAGO26K-906* and *DB111K-174*, which are extracted from YAGO [MBS14] and DBpedia [LIJ15] respectively.

We use YAGO26K-906 and DB111K-174, which are extracted from the connected subsets of YAGO [MBS14] and DBpedia [LIJ15] respectively, for experimental purpose. The datasets are constructed through the following steps:

1. We first filter out all attribute triples, since such triples do not represent the relations of entities or concepts. After randomly sample some relational triples from the rest of the filtered dataset since original YAGO and DBpedia both have large collections of instance-view triples.
2. After we obtain the entity set of instance view, we extract cross-view alignment of those entities to the ontology view of the two KBs. As a result, a portion of entities are linked to the associated concepts, which are naturally the nodes in the ontology view.
3. Given all the associated concepts from step (2), we construct the corresponding ontology views base on the intersecting subgraph of the original ontologies.

It is noteworthy that the original YAGO has a taxonomical ontology with only three types of

Table 2.1: Statistics of datasets.

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

semantic relations, which casts limitation on semantic relations among concepts. Therefore, we enrich the ontology view of YAGO using the knowledge from ConceptNet [SCH17], another KB which contains a large collection of meta-relations among concepts. The concepts in ConceptNet and YAGO are easily aligned by the shared WordNet-based IDs or concept names. Consequently, we obtain two datasets that are much larger than FB15K – the widely adopted instance KG benchmark dataset by many recent works [BUG13, YYH15, LLS15, NRP16].

Table 2.1 provides the statistics of both datasets. Normally, the instance-view KG is significantly larger than the ontology-view graph. Also, we notice that the two KBs are different in the density of type links, i.e., DB111K-174 has a much higher entity-to-concept ratio (643.4) than YAGO26K-906 (28.7).

Datasets are available at <https://github.com/JunhengH/joie-kdd19>.

2.4.2 KG Triple Completion

The objective of triple completion is to construct the missing relation facts in a KG structure, which directly tests the quality of learned embeddings. In our experiment, this task spans into two sub-tasks for instance-view KG 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 [BUG13] for instance-only KG embeddings. Each JOIE variant is trained on $\mathcal{G}_I^{\text{train}}$ and $\mathcal{G}_O^{\text{train}}$ triples along with all cross-view 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 intra-view 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 [BUG13, YYH15].

As for the hyperparameters in training, we select the dimensionality option 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 JOIE with TransE, DistMult and HolE as well as TransC [LHL18]. 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 2.2; (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 cross-view links by adding one additional relation “*type_of*” to them, denoted as (*all*) in Table 2.2. (iii) TransC, trained on both views of a KB, 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.

Table 2.2: Results of KG triple completion. H@1 and H@10 denote *Hit@1* and *Hit@10* respectively. For each group of model variants with the same intra-view model, the best results are bold-faced. The overall best results on each dataset are underscored.

Graphs	\mathcal{G}_O KG Completion			\mathcal{G}_I KG Completion		
Metrics	MRR	H@1	H@10	MRR	H@1	H@10
TransE (base)	0.195	14.09	34.51	0.145	12.29	20.59
TransE (all)	0.187	13.73	35.05	0.189	14.72	24.36
TransC	0.252	15.71	37.79	–	–	–
JOIE-TransE-CG	0.264	16.38	35.45	0.189	11.16	29.44
JOIE-TransE-CT	0.292	18.72	44.14	0.240	14.49	33.47
JOIE-HATransE-CT	0.306	18.62	51.72	0.263	16.72	38.46
DistMult (base)	0.253	22.91	28.76	0.197	17.72	25.08
DistMult (all)	0.288	24.06	31.24	0.156	14.32	16.54
JOIE-Mult-CG	0.274	18.80	37.45	0.198	11.16	27.91
JOIE-Mult-CT	0.309	20.40	46.15	0.207	14.71	30.43
JOIE-HAMult-CT	0.296	19.39	45.48	0.202	13.72	31.10
HolE (base)	0.265	25.90	28.31	0.192	18.70	20.29
HolE (all)	0.252	24.22	26.56	0.138	11.29	14.43
JOIE-HolE-CG	0.253	18.75	34.11	0.167	13.04	22.33
JOIE-HolE-CT	0.313	20.40	47.80	0.229	20.85	28.42
JOIE-HAHolE-CT	0.327	22.42	52.41	0.236	16.72	30.96

(a) KG triple completion on YAGO26K-906.

Graphs	\mathcal{G}_O KG Completion			\mathcal{G}_I KG Completion		
Metrics	MRR	H@1	H@10	MRR	H@1	H@10
TransE (base)	0.327	22.26	49.01	0.313	23.22	46.91
TransE (all)	0.318	22.70	48.12	0.539	47.90	61.84
TransC	0.359	24.83	49.31	–	–	–
JOIE-TransE-CG	0.394	27.75	51.20	0.598	53.84	71.79
JOIE-TransE-CT	0.443	32.10	67.89	0.622	58.10	72.97
JOIE-HATransE-CT	0.473	33.79	71.37	0.591	52.07	79.65
DistMult (base)	0.265	25.95	27.63	0.235	15.18	29.11
DistMult (all)	0.280	27.24	29.70	0.501	45.52	64.73
JOIE-Mult-CG	0.320	23.44	49.49	0.532	46.15	68.91
JOIE-Mult-CT	0.404	26.55	60.86	0.563	50.50	71.62
JOIE-HAMult-CT	0.369	24.82	55.86	0.521	38.46	77.25
HolE (base)	0.301	29.24	31.51	0.227	18.91	32.83
HolE (all)	0.295	28.70	30.32	0.432	38.80	56.05
JOIE-HolE-CG	0.361	24.13	46.15	0.469	41.89	62.16
JOIE-HolE-CT	0.425	29.09	66.88	0.514	43.24	69.23
JOIE-HAHolE-CT	0.464	33.11	69.56	0.503	40.80	71.03

(b) KG triple completion on DB111K-174.

Results As reported in Table 2.2, we categorize the results into three different groups based on the intra-view models. Though three intra-view 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 intra-view models, translation based models performs better than similarity based models on ontology population and instance-view KG completion on the DB111K-174 dataset. This is because these graphs are sparse, and TransE is less hampered by the sparsity in comparison to the similarity-based techniques [PAG17]. By applying the HA technique in the intra-view models with CT, the performance on instance-view triple completion is noticeably improved in most cases in comparison to the default intra-view CT-based models, especially in variants with translation and circular correlation based intra-view models.

Generally, JOIE provides an effective method to train two-view KB 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.

2.4.3 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 cross-view 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 cross-view links $\mathcal{S}^{\text{train}}$. Hyperparameters are carried forward from the triple completion task, in order to evaluate

Table 2.3: Results of entity typing on YAGO26K-906 and DB111K-174.

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

under controlled variables. In the test phase, given a specific entity e_q , we rank the concepts based on their embedding distances from 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 cross-view links (e, c) to triples $(e, r_T = \text{“type_of”}, c)$. 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 [CTY17]) in their model and test with distance-based ranking.

Results Results are reported in Table 2.3. 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 cross-view 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 intra-view model and CT as the cross-view association 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 intra-view models, translation based intra-view model better differentiates between different entities and different concepts in KGs with directed relations and meta-relations in the KB [PAG17]. 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 dimensionalities, 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 intra-view models, it is also observed that HA intra-view 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 intra-view 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 structures³. Comparing the two datasets, our experiments show that, JOIE generally achieves similar accuracy 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 2.4.4 and Section 2.4.5, we provide more experiments and insights on the benefits of representation learning with JOIE.

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

2.4.4 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 triple completion process can already populate the ontology view with seen meta-relations, by answering the query like (“*Concert*”, “*Related to*”, *?t*) in the KG 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. A similar example with ontology inference between “*Computer Scientist*” and “*University*” is shown in Figure 2.4. 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*”).

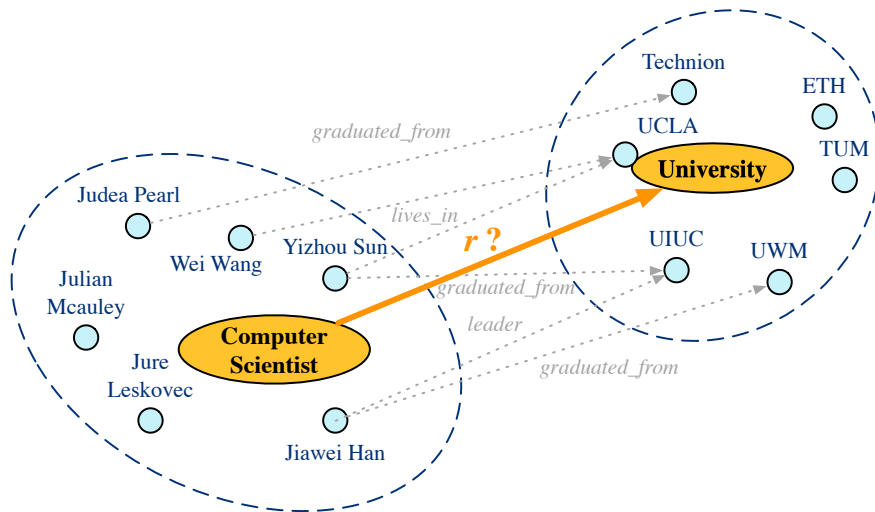


Figure 2.4: Intuition on how the two-view KG can help with ontology population, i.e. to infer potential new relations between concepts.

Our proposed JOIE provides a feasible solution by leveraging the cross-view association

model that bridges the two views of the KG, 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 intra-view 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 2.5 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.

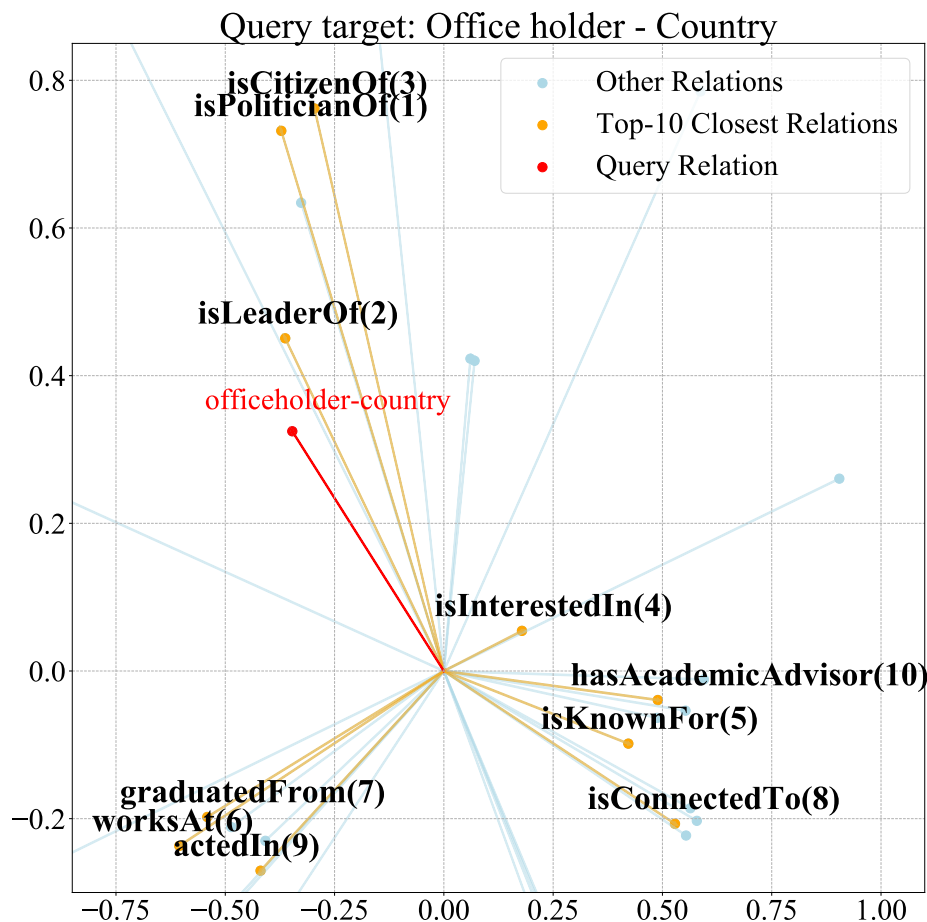


Figure 2.5: Examples of ontology population by finding the closest relations in the instance view for the query “Office Holder-Country”. Top 10 predicted relations are plotted with their ranks.

Table 2.4: 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, <i>isKnownFor</i> , university (1.098) scientist, <i>created</i> , 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, <i>graduatedFrom</i> , club (1.497) boxer, <i>isConnectedTo</i> , club (1.552)
(TV station, ?r, country)	TV station, headquarter , country (1.221) TV station, <i>parentOrganisation</i> , country (1.246) TV station, <i>appointer</i> , country (1.253) TV station, broadcastArea , country (1.266) TV station, principalArea , country (1.271)
(scientist, ?r, scientist)	scientist, <i>deputy</i> , scientist (0.204) scientist, doctoralAdvisor , scientist (0.218) scientist, doctoralStudent , scientist (0.221) scientist, relative , scientist (0.228) scientist, spouse , scientist (0.230)

Table 2.4 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. From these top predictions, we observe that most populated ontology triples migrated from the instance view are meaningful.

Long-tail entity typing In KGs, the frequency of entities and relations often follow a long-tail distribution (Zipf’s law) in both YAGO26K-906 and DB111K-174 datasets, which is confirmed by the histogram in Figure 2.6. As shown in Figure 2.6a and Figure 2.6b, both YAGO26K-906 and DB111K-174 discover such a property. Over 75% of total entities has less than 15 occurrences. Those long-tails entities, types and relations are difficult for representation learning algorithms to capture due to being few-shot in training cases.

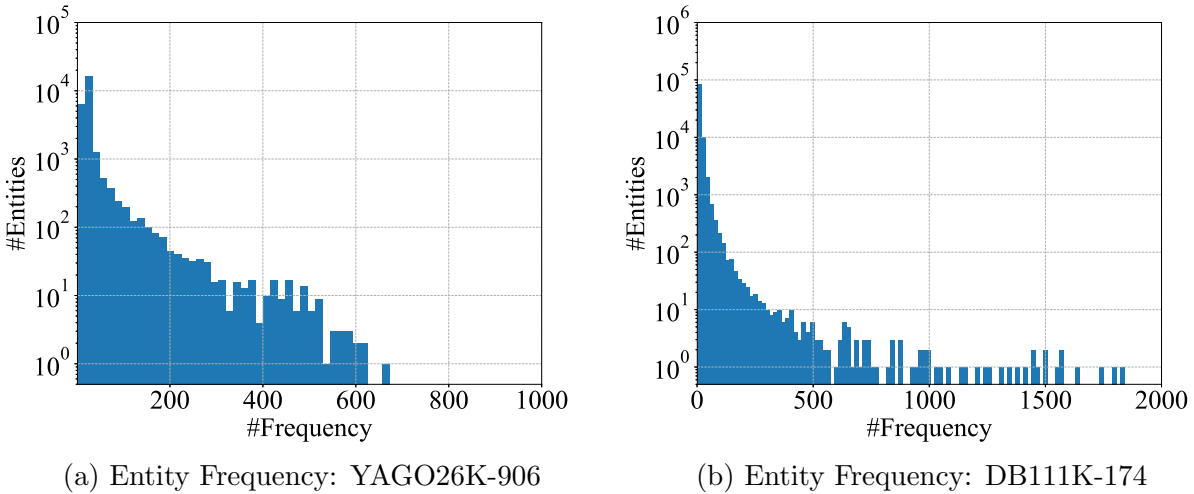


Figure 2.6: Long-tail distribution holds on entity frequency from both YAGO26K-906(a) and DB111K-174(b)

In this case study, we select the entities with considerably low frequency⁴, which involve around 15%-30% of total entities in the instance view of the two KB datasets. Then, we evaluate the entity typing task for these long-tail entities. Table 2.5 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 2.4.3, 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 2.6, 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 instance-view graph and ontology-view graph are relatively sparser. JOIE is still

⁴In this experiment, we select entities in YAGO26K-906 which occurs less than 8 times and entities in DB111K-174 which occurs less than 3 times.

Table 2.5: 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 2.6: 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

able to make correct predictions of low-frequency entities while other baselines models can only output inaccurate predictions.

2.4.5 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 cross-view links in training, and the effect of adopting negative sampling in cross-view association models.

Dimensionality Dimensionality is a key hyperparameter that affects the quality of the obtained embeddings. Figure 2.7a shows the *MRR* of model variants with the CG-based cross-view association according to different embedding dimensions d . It is observed in Figure 2.7a 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 $d = 300$.

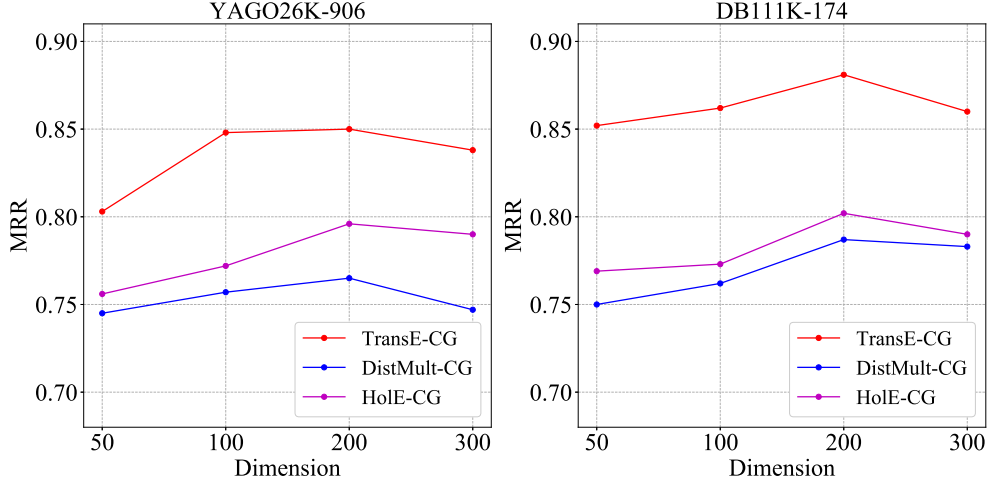
Similarly we plot *MRR* scores for both dataset with CT model variants in Figure 2.7b. We compare four different dimensionality settings of (d_1, d_2) : $(100, 20)$, $(100, 50)$, $(300, 50)$ and $(300, 100)$ ⁵. Most of the JOIE variants achieve their best performance under the embedding setting $(d_1, d_2) = (300, 50)$ 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 KG, as well as inefficient training and prediction processes.

Sufficiency of Type Information Cross-view 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 cross-view links and their sufficiency in training.

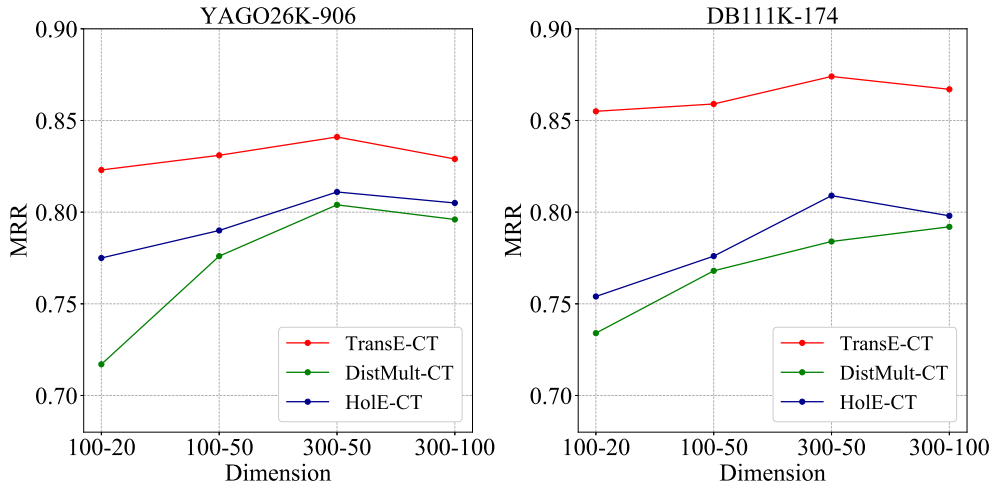
We define the train set ratio $\nu = \{0.2, 0.4, 0.6, 0.8\}$, which means the proportions of the cross-view links that are used for training JOIE. *MRR* score is reported in Figure 2.8a on YAGO26K-906 and Figure 2.8b on DB111K-174. As expected, when the proportion of cross-view links used for training 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% cross-view links still outperforms MTransE trained with 60% cross-view 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 KG 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.

⁵ $(d_1, d_2) = (100, 20)$ denotes that entities are embedded with $d_1 = 100$ dimensional vectors and concepts are embedded with $d_2 = 20$ dimensional vectors



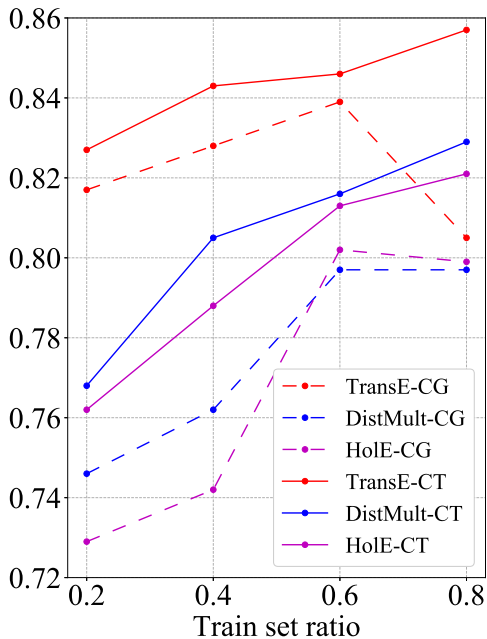
(a) Different dimensions with CG variants



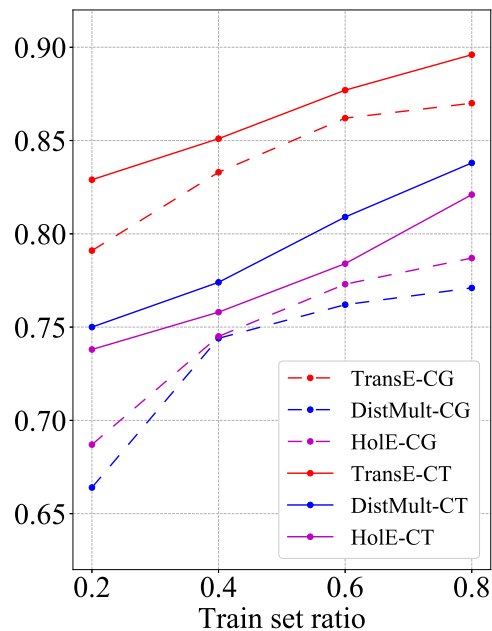
(b) Different dimensions with CT variants

Figure 2.7: Performances of entity typing task on both datasets with different entity and concept embedding dimensionalities

Effects of Negative Sampling Negative sampling is widely applied in the encoding process of a single KG structure [BUG13, YYH15]. One interesting question is whether to use negative sampling for capturing the cross-view 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 2.7. It is our finding that there is a significant performance drop if negative sampling is disabled in CT, while negative sampling has less effect on CG. We hypothesize that the



(a) YAGO26K-906

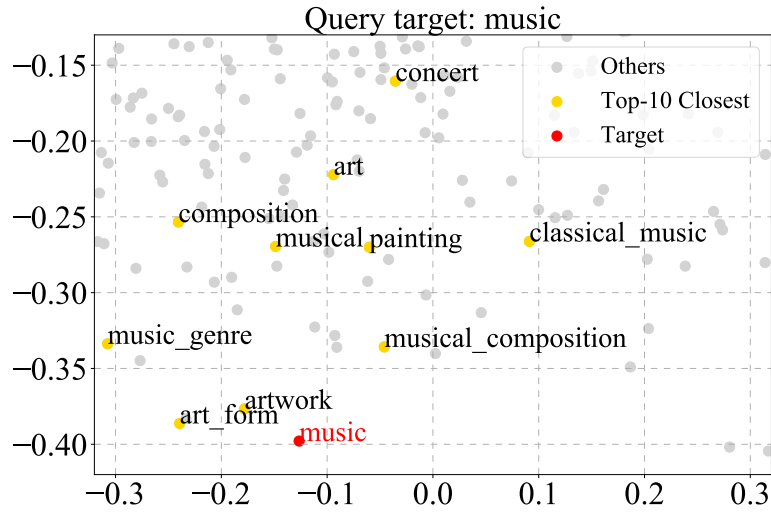


(b) DB111K-174

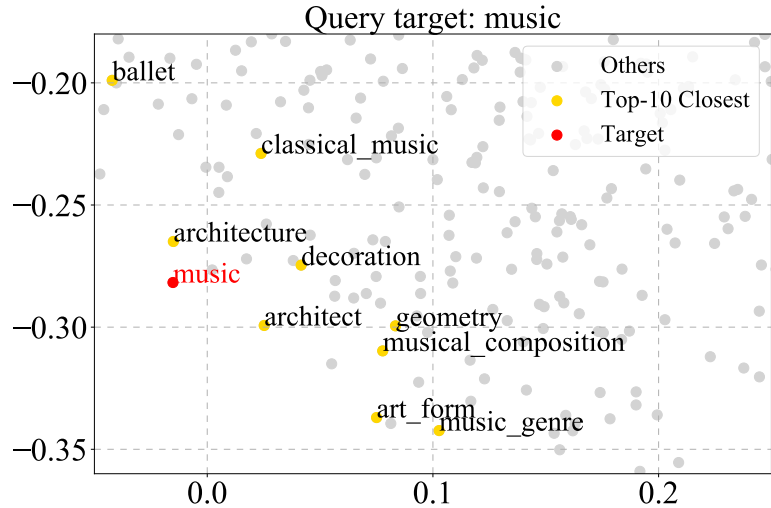
Figure 2.8: The effect of training the model using different proportions of cross-view links on (a) YAGO26K-906 and (b) DB111K-174

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 2.9. 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.



(a) JOIE, With negative sampling



(b) JOIE, without negative sampling

Figure 2.9: Visualize effects on embeddings of negative sampling on cross-view links

2.5 Conclusion

In this chapter, we propose a novel model JOIE aiming to jointly embed real-world entities and ontological concepts. We characterize a two-view knowledge base. In the embedding space, our approach jointly captures both structured knowledge of each view, and cross-view links that bridges the two views. Extensive experiments on the tasks of KG completion and

Table 2.7: 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

entity typing show that our model JOIE can successfully capture latent features from both views in KBs, and outperforms various state-of-the-art baselines.

We also point out future directions and improvements. The formulation of a two-view knowledge graph remains a simplification of real-world complex and hierarchical structures. In later chapters (Chapter 3, Chapter 4 and Chapter 5), we continue to explore more approaches to model ontologies in bioinformatics (gene ontology, disease ontology, etc) and e-commerce (product ontologies).

CHAPTER 3

Bio-JOIE: Joint Representation Learning of Biological Knowledge Bases

3.1 Introduction

The outbreak of COVID-19 (Coronavirus Disease-2019) has infected millions of people and caused high death tolls since the end of 2019, as worldwide social and economic disruption. Tremendous efforts have been made to discover the infection mechanism of the causative agent, named SARS-CoV-2. One important and urgent task is to understand the mechanism in which viral proteins interact with human proteins. The new findings will enrich the annotation of viral genomes [GJB20] in biomedical knowledge bases (KBs). Constructing and populating such biomedical KBs can significantly improve our understanding of the processes by which SARS-CoV-2 affects different cells in human body and will serve as the foundation for many important downstream applications such as vaccine development [KSS19], drug repurposing [ZHS20, GJB20] and drug side effect detection [ZAL18].

In general, biological KBs, often stored as knowledge graphs (KGs), consist of various biological entities, their properties and relations. These KBs can be categorized in different domains, such as gene annotation, functional proteomic analysis, and transcriptomic profiling. Specifically, gene ontology (GO) [Con18, HSM15] is the most widely used resource for gene function annotation; STRING [SMC16], PDB [BHN07] and neXtProt [LAB12] collect the knowledge accumulated from functional proteomic analysis; Expression Atlas [PMM20] is a database facilitating the retrieval and analysis of gene expression studies. While those KBs provide the essential sources of knowledge for *in silico* research in the corresponding domains,

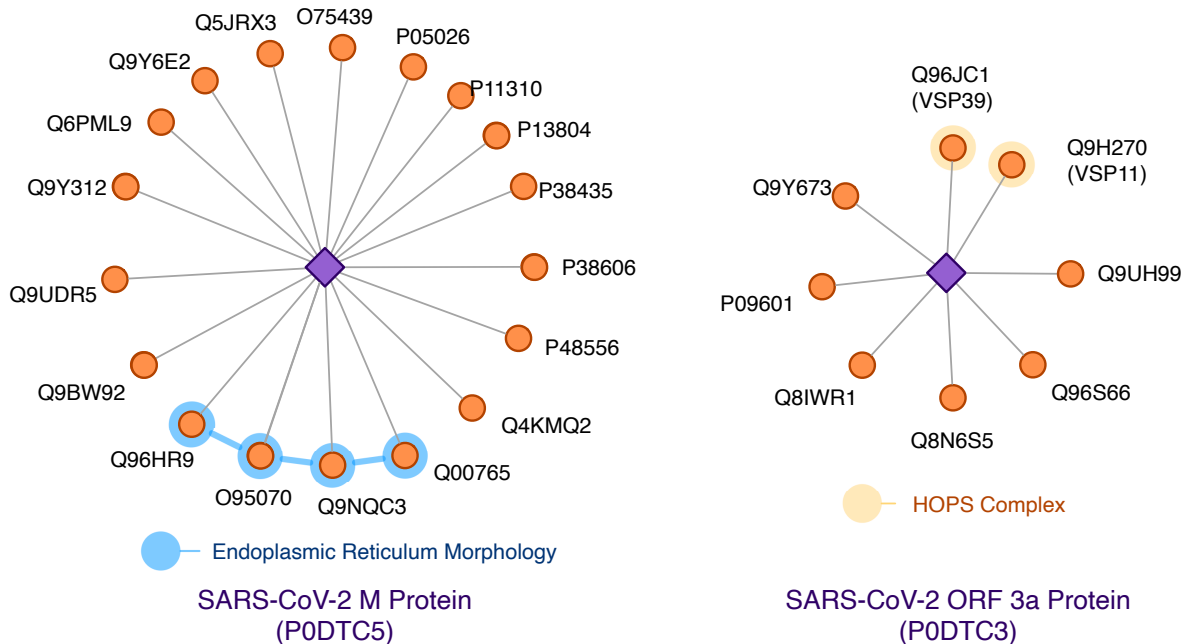


Figure 3.1: Two examples of SARS-CoV-2-human protein interactions: M protein (left) and ORF3a protein (right). The purple diamonds refer to the viral proteins and the orange circles refer to the high-confidence human protein target. Proteins highlighted in blue are involved in certain biological processes, and proteins highlighted in yellow are arranged in a protein complex.

such domain-specific knowledge is often sparse and costly to apprehend [MHR19, TWM12]. For example, PPI networks can be far from complete given the information supported by experimental results or suggested by computational inference [HLW18, MHR19]. (author?) [MHR19] indicate that the numbers of PPIs in BIOGRID [OSB19] for non-model organisms are far less than expected, specifically, there are only 107 interactions for tomato (*Solanum lycopersicum*) and 80 interactions for pig (*Sus scrofa*). Evidently, relying on the KG from a single domain presents the risk of learning from limited and scarce information.

The stored knowledge is often interrelated across different perspectives. Hence, the missing knowledge in certain KBs can be transferred from other KBs, and thus provide a more comprehensive representation of the biological entities. Taking the protein-protein interaction (PPI) examples of the new SARS-CoV-2 proteins as illustrated in Figure 3.1, SARS-CoV-2 M protein interacts with a list of human proteins, and five of them are involved in the endoplasmic reticulum (ER) morphology process as suggested by the gene ontology

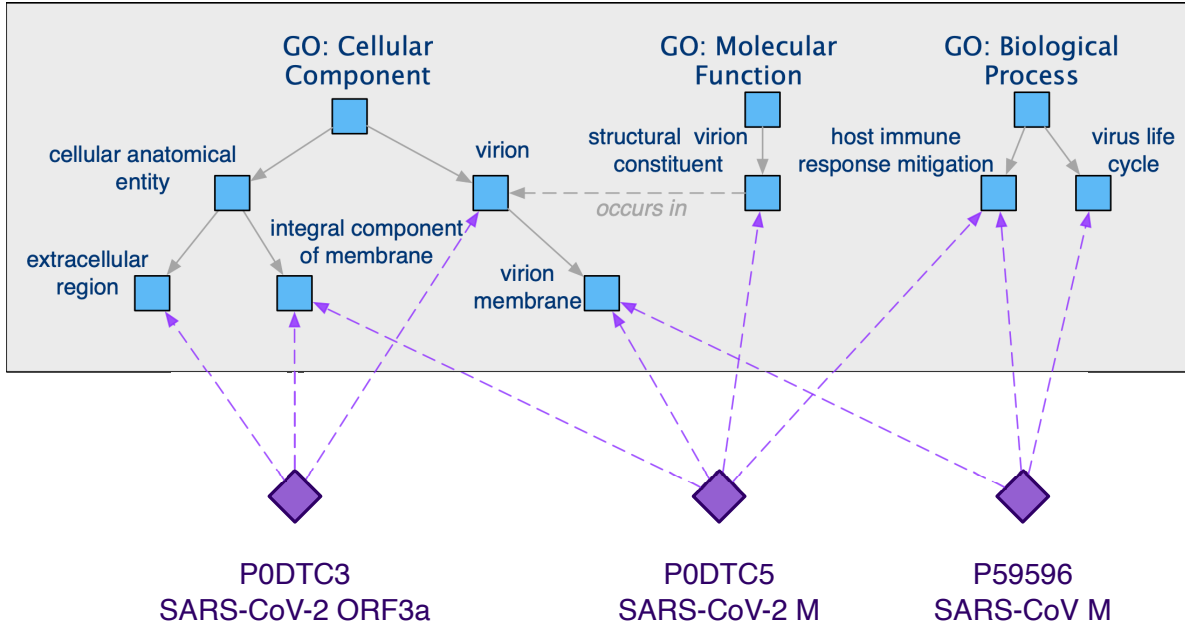


Figure 3.2: Examples of gene ontology annotation enrichment on three representative SARS-CoV or SARS-CoV-2 proteins, which possess multiple properties across three biological aspects: biological processes, cellular components and molecular functions.

annotation (GO:0005783). Similarly, the SARS-CoV-2 ORF3a also interacts with a list of human proteins. Among these proteins, VSP39 and VSP11 are the core subunits of HOPS complex, presenting a binding action as suggested by the STRING database. While aligning the gene ontology annotations of the SARS-CoV-2 M protein as demonstrated in Figure 3.2, the SARS-CoV M protein presents a similar set of gene ontology annotations, such as “host immune mitigation” and “virion membrane”, suggesting that the side knowledge of gene ontology annotations can facilitate the inference of interactions for related proteins. More generally, the sparse domain information can always benefit from the supplementary knowledge from other relevant domains, therefore calling upon a plausible method to support the fusion and transfer of knowledge across multiple biological domains.

Regardless of the importance and advantages of knowledge fusion across different domains [BN09, BB14], fewer efforts have been devoted to incorporating knowledge from different domains for a specific task in computational biology studies. Onto2vec [SGH18] presents one state-of-the-art learning approach that successfully bridges gene ontology annotations with the protein representation. However, the known PPI information is neglected and not

encoded in the obtained protein embeddings.

To combine multiple domain-specific biological knowledge, and facilitate knowledge transfer across different domains, we propose **Bio-JOIE**, a **JoInt Embedding** learning framework for multiple domains of **Biological** KBs. In **Bio-JOIE**, two model components are jointly learned, i.e., a knowledge model characterizes different domain-specific KGs in separate low-dimensional embedding spaces, and a transfer model captures the cross-domain knowledge association. More specifically, the knowledge model encodes the relational facts of entities in each view into the corresponding embedding space separately, with a hierarchy-aware technique designated for the hierarchically-layered domains. Besides, the transfer model seeks to transfer the knowledge between pairs of domains by employing a weighted non-linear transformation across their embedding spaces. In evaluation, we apply the **Bio-JOIE** on several PPI networks with Gene Ontology annotations and the entire gene ontology and evaluate by PPI predictions. We compare **Bio-JOIE** with that of the state-of-the-art representation learning approaches on multiple species, including SARS-CoV-2-Human PPIs, with different model settings. Our best **Bio-JOIE** outperforms alternative approaches by 7.4% in PPI prediction.

Our contributions are 4-fold.

- First, we construct a general framework for learning representations across different domain-specific KBs, including the dynamically changing SARS-CoV-2 KB.
- Second, we emphasize and demonstrate that cross-domain representation learning by the proposed **Bio-JOIE** can improve the inference in one domain by leveraging the complementary knowledge from another domain. Extensive experiments on different species confirm the effectiveness of cross-domain representation learning.
- Third, **Bio-JOIE** also demonstrates cross-species transferability to improve PPI predictions among multiple species by knowledge population from gene ontology.
- Fourth, the protein representations learned from **Bio-JOIE** can be leveraged for different tasks. Specifically, we show that the protein embeddings trained on PPI network

and gene ontology present the potential to better group enzymes into different enzyme commission families. Tremendous efforts have been made to discover the infection mechanism of the causative agent, named SARS-CoV-2.

3.2 Related Work

In the past decade, much attention has been paid to representation learning of KBs. Methods along this line of research typically encode entities into low dimensional embedding spaces, where the relational inference [WZF14], proximity measures and alignment [CTY17] of those entities can be supported in the form of vector algebras. Therefore, they provide efficient and versatile methods to incorporate the symbolic knowledge of KGs into statistical learning and inference. Some existing approaches focus specifically on computational biology studies [AKM17, SGH18, CJZ19, YCH15, HYG15], which similarly embed features of biological entities within low-dimensional representations. One representative work related to ours is Onto2Vec [SGH18], in which protein representations are learned by incorporating the full semantic content of gene ontology in the feature learning using Word2Vec [MSC13]. However, Onto2Vec relies on the ontology information, which falls short of capturing the multi-relational semantic facts that are important to characterize the proximity of biological entities. For example, regarding the protein and GO terms, the PPI knowledge and the non-hierarchical relationships between gene ontology entities (such as “regulates”) are not considered.

Another thread of related work is joint representation learning for multiple KGs, where embedding models are learned to bridge multiple relational structures for tasks such as entity alignment and type inference. MTransE [CTY17] jointly learns a transformation across two separate translational embedding spaces based on one-to-one seed alignment of entities. Later extensions of this model family, such as KDCoE [CTC18a], MultiKE [ZSH19] and JAPE [SHL17], require additional information of literal descriptions [CTC18a] and numerical attributes of entities [SHL17, TQZ19, ZSH19] that are generally not available for biological KB. Our recent development in this line of research, i.e. JOIE [HCY19] learns a many-to-

one mapping between entity embeddings and ontological concept embeddings, and aims at resolving the entity type inference task using the latent space of the type ontology. One of the caveats is that JOIE does not specifically incorporate the specificity of concepts in the ontology in the transfer process, which we find to be particularly beneficial in this problem setting. Besides, the aforementioned methods are mostly for general encyclopedia KBs (such as Wikidata, and DBpedia) and have not been adapted for the purpose the modeling biological KBs. More specifically, in contrast to these methods, our method features the characterization of more complicated many-to-many associations between proteins and GO terms. Besides, instead of predicting the alignment of entities, we focus on transferring relational knowledge from one domain to enhance the prediction on the other.

Also, regarding the task of this chapter, predicting protein-protein interactions (PPIs) and characterizing the interaction types are one of the essential tasks in computational biology. In the past decade, a wide selection of research works to address the PPI prediction problem. Representative examples are homology-based methods [POK16], which map a pair of sequences to known interacting proteins by BLAST, and sequence-based statistical learning models [GYW08, YCH15], which rely on extracting protein sequences as primary features. Such sequential features include CT [SZL17], CTD [DSH17], multi-scale continuous and discontinuous (MCD) descriptors [YZZ14], and local phase quantization (LPQ) [WYL15]. Some recent works also utilize alternative deep learning based techniques [SZL17], including stacked autoencoders (SAE) [WYL17], convolutional neural networks (CNN) [LGY18] and Siamese residual RCNN [CJZ19]. Other than sequence based methods, some network factorization based methods have also been proposed between drug and target proteins [ZLW22].

3.3 Materials and Method

In this section, we present the proposed method to support representation learning and cross-domain knowledge transfer on biological KBs. Without loss of generality and aligned with the evaluation of the proposed Bio-JOIE, we refer to two domain-specific KGs in the following section to PPI networks and the gene ontology graph. We begin with the formalized

descriptions of the materials and tasks.

3.3.1 Preliminary

Materials. A typical biological KB can be viewed as relational data that are presented as an edge-labeled directed graph \mathcal{G} , which is formed with a set of entities (e.g. proteins) \mathcal{E} and a set of relations (e.g. interaction types) \mathcal{R} . A triple $(s, r, t) \in \mathcal{G}$ represents a $r \in \mathcal{R}$ typed relation between the source and target entities $s, t \in \mathcal{E}$. As stated, we continue with the modeling on KGs of two domains, PPI and gene ontology. For example, in the PPI network, a triple (FBgn0011606, binding, FBgn0260855) simply states the fact that two proteins (from fly) have binding interaction; and in gene ontology, a triple (GO:0008152, is_a, GO:0008150) similarly represents that GO:0008152 (a unique identifier of “metabolic process”) is one subclass of GO:0008150 (a unique identifier of “biological process”). Our model seeks to capture the protein information in the triples (s_p, r_p, t_p) of PPI graph \mathcal{G}_p in a k_p -dimensional embedding space, where we use boldfaced notations such as $\mathbf{s}_p, \mathbf{r}_p, \mathbf{t}_p \in \mathbb{R}^{k_p}$ to denote the embedding representation. Similarly, gene ontology is another graph \mathcal{G}_o formed with a set of GO terms \mathcal{E}_o and a set of semantic relations \mathcal{R}_o . The triple $(s_o, r_o, t_o) \in \mathcal{G}_o$ identifies a semantic relation of GO terms, while we also observe hierarchical substructures formed by “subclass” or “is_a” relation as the aforementioned example. The gene ontology is embedded in another space \mathbb{R}^{k_o} , such that k_p and k_o may not be equivalent. We use $(o, p) \in \mathcal{A}$ to denote a *GO term annotation* where a GO term $o \in \mathcal{E}_o$ describes a protein $p \in \mathcal{E}_p$ of its corresponding functionality, and \mathcal{A} denotes the set of such associations. As introduced in Section 3.1, we consider SARS-CoV-2-Human interaction as a similar (but significantly smaller) KBs with the same structures as \mathcal{G}_p , which serves as an extension of human PPI networks.

Tasks. To validate the learned embedding of biological entities (proteins and GO terms in this context), we address the following two tasks. (i) *PPI type prediction* aims at predicting the interaction type between two interacting proteins, including *SARS-CoV-2 related PPIs*; (ii) *Protein clustering and family identification* aims at clustering the existing proteins and

helps identify the clusters based on Enzyme Commission (EC) numbers.

Methods. The model architecture of Bio-JOIE is shown in Figure 3.3. The proposed Bio-JOIE jointly learns two types of model components to connect the two views of structured knowledge. Knowledge models are responsible for representing the relational knowledge of PPI and that of GO term into two separate embedding spaces \mathbb{R}^{k_p} and \mathbb{R}^{k_o} by using KG embedding and hierarchy-aware regularization. On top of that, a transfer model learns a transformation to connect between the representations of GO term relation facts and PPI based on partially provided GO term assignments. In particular, we investigate weighted *transfer techniques* to better capture the knowledge transfer, for which the weights reflect the specificity of the assigned GO term to a protein. The following of this section describes the model components and the learning objective of Bio-JOIE in detail.

3.3.2 Knowledge Model

The knowledge models seek to characterize the semantic relations of GO terms and PPI information into separate embedding spaces. In each embedding space, the inference of relations or interactions is modeled as specific algebraic vector operations. As mentioned, the two views of gene ontology and PPI are embedded to separate embedding spaces.

To capture a triple (s, r, t) from either of the two domains, a cost function $f_r(s, t)$ is provided to measure its plausibility. A lower score indicates a more plausible triple. We can adopt multiple vector operations in the defined embedding space with three representative examples defined as follows, i.e. translations (TransE [BUG13]), Hadamard product [YYH15] and circular correlation (HolE [NRP16]). The cost functions are given as follows, where the symbol \circ denotes Hadamard product, and $\star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes circular correlation defined as $[\mathbf{a} \star \mathbf{b}]_k = \sum_{i=0}^d a_i b_{(k+i) \bmod d}$.

$$\begin{aligned}
 f_r^{\text{Trans}}(\mathbf{s}, \mathbf{t}) &= \|\mathbf{s} + \mathbf{r} - \mathbf{t}\|_2 \\
 f_r^{\text{Mult}}(\mathbf{s}, \mathbf{t}) &= -(\mathbf{s} \circ \mathbf{t}) \cdot \mathbf{r} \\
 f_r^{\text{HolE}}(\mathbf{s}, \mathbf{t}) &= -(\mathbf{s} \star \mathbf{t}) \cdot \mathbf{r}
 \end{aligned}$$

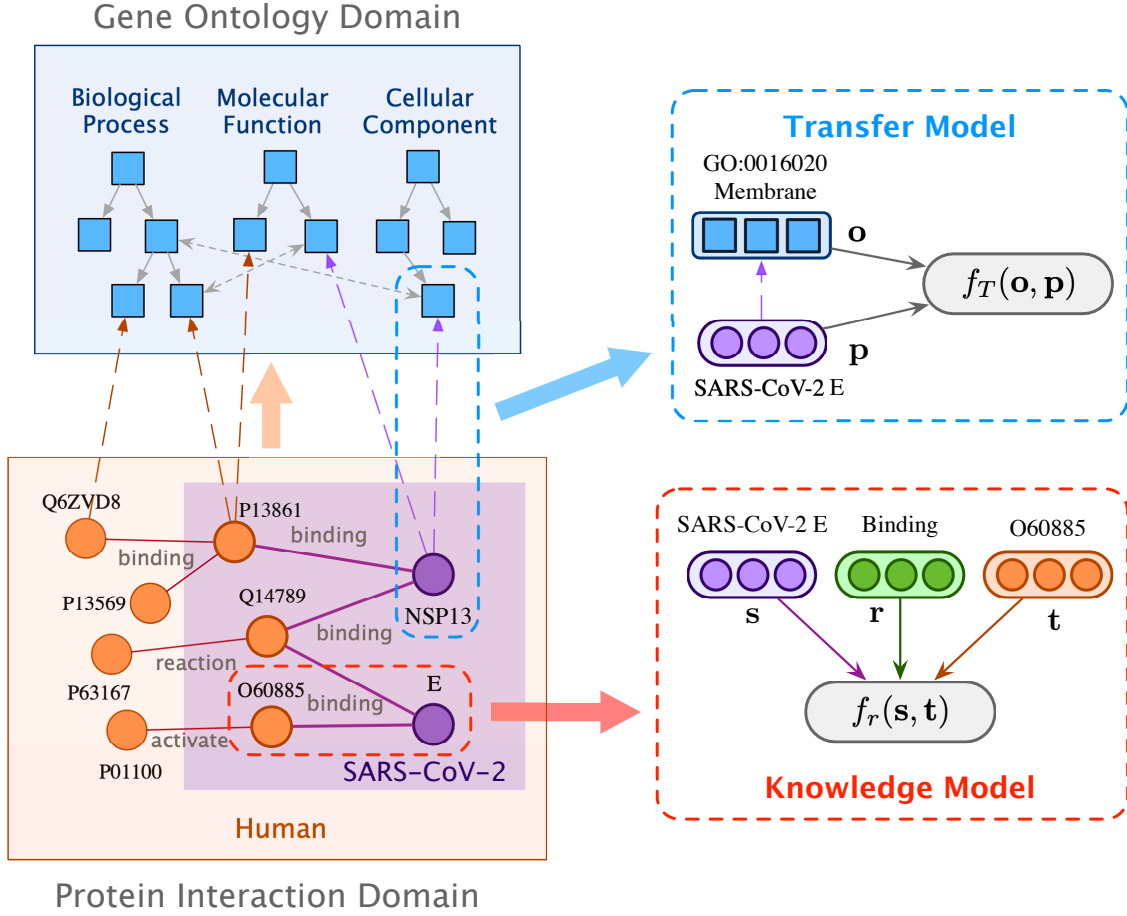


Figure 3.3: Model architecture of Bio-JOIE. The Knowledge Model seeks to encode relational facts in each domain respectively (such as proteins and gene ontology). Meanwhile, the Transfer Model learns to connect both domains and enable knowledge transfer across protein and gene ontology.

Since most of the relations in PPI networks are symmetric (such as binding and catalysis), we apply the Hadamard product based function. The learning objective of a knowledge model on a graph G is to minimize the following margin ranking loss,

$$\mathcal{L}_K^G = \frac{1}{|\mathcal{G}|} \sum_{(s,r,t) \in \mathcal{G}} \max \{ f_r(s, t) + \gamma^G - f_r(s', t'), 0 \}$$

where γ^G is a positive margin, and a negative sample $(s', r, t') \notin \mathcal{G}$ is created by randomly substituting either s or t using Bernoulli negative sampling [WZF14]. With regard to the two domains of relational knowledge (proteins and gene ontology) \mathcal{G}_p and \mathcal{G}_o , we denote the

learning objective losses as $\mathcal{L}_K^{\mathcal{G}_p}$ and $\mathcal{L}_K^{\mathcal{G}_o}$.

Hierarchy-aware Encoding Regularization As mentioned in Section 3.3.1, it is observed that some ontological knowledge can form hierarchies [CTC18b], which is typically constituted by a relation with the implicit hierarchical property, such as “subclass_of”, as substructures. In gene ontology, more than 50% of the triples have such relations. To better characterize such hierarchies, we model such substructures differently from the aforementioned DistMult and many others by adding hierarchy regularization. More specifically, given entity pairs $(e_l, e_h) \in S$ where e_l is a subclass of e_h , we model such hierarchies by minimizing the distance between coarser concepts and associated finer concepts in embedding space. Hence, the loss is simply defined as

$$\mathcal{L}_{(\text{HA})} = \frac{1}{|S|} \sum_{(e_l, e_h) \in S} [||e_l - e_h||_2 - \gamma_{\text{HA}}]_+$$

where $[x]_+ = \max\{x, 0\}$ and γ_{HA} is also a positive margin parameter. This penalizes the case where the embedding of e_l falls out the γ_{HA} -radius neighborhood centered at the embedding of e_h .

Relation Inference Given the learned embeddings and a pair of query proteins $((p_1, p_2))$, we can predict the most plausible interaction type r by selecting the optimal $f_r(p_1, p_2)$ score. We can also provide predictions for possible protein targets given the query of the subject protein and specific interaction type $(p, r, ?t)$ by populating the selection proteins with top score $f_r(p, t)$ from the knowledge model. Details about each task are curated in Section 3.4.3 and 3.4.5.

3.3.3 Transfer Model

The transfer model learns to connect between the above two relational embedding spaces via a non-linear transformation. The transformation is induced based on the GO term assignments, towards the goal to collocate the associated GO terms and proteins in an

embedding space after transformation. Hence, the affinity of embedding structures of gene ontology and PPIs can be captured. This allows the relational knowledge to transfer across and complement the learning and inference on both domains.

Given each GO term assignment $(o, p) \in \mathcal{A}$, following function $f_T(o, p)$ measures the plausibility of the transformation that is favored to be minimized.

$$f_T(o, p) = \|\sigma(\mathbf{M}_T \cdot \mathbf{p} + \mathbf{b}_T) - \mathbf{o}\|_2$$

$\mathbf{M}_T \in \mathbb{R}^{k_o \times k_p}$ thereof is a weight matrix and $\mathbf{b}_T \in \mathbb{R}^{k_p}$ is a bias vector. σ is either the identify function, or a non-linear function as tanh, the latter thereof aims at smoothing the transformation with additional non-linearity.

3.3.3.1 Basic Transfer Model

The basic strategy to learn the transfer model is to treat each GO term assignment evenly, and thereby minimizing the following learning objective loss.

$$\mathcal{L}_{T_1} = \frac{1}{|\mathcal{A}|} \sum_{(o,p) \in \mathcal{A}} \max \{f_{T_1}(o, p) + \gamma^{\mathcal{A}} - f_{T_1}(o', p'), 0\}$$

$(p', o') \notin \mathcal{A}$ thereof is a negative sample by randomly substituting p' , and $\gamma^{\mathcal{A}}$ is a positive margin.

3.3.3.2 Weighed Transfer Model

Since some ontological knowledge, such as gene ontology, may form hierarchical structures, where GO terms in lower levels typically describe more specified gene functionality. During the characterization of associations between GO terms and proteins, in contrast to general GO terms, more specified GO terms necessarily carry more precise descriptions of the proteins. Hence, an improved transfer model weights among GO term associations to a protein for the purpose of more attentively capturing those with more specific GO terms. Let $\omega(o)$

be the weight is specifically assigned to o , the objective of the weighted transfer model is to minimize the following loss,

$$\mathcal{L}_{T_2} = \frac{1}{|\mathcal{A}|} \sum_{(o,p) \in \mathcal{A}} \max \left\{ \frac{\omega(o)}{C} [f_{T_2}(o,p) + \gamma^A - f_{T_2}(o',p')], 0 \right\}$$

where C is a normalizing constant to constrain that $\sum_{(o,p)} \frac{\omega(o)}{C} = 1$ for a specific protein \hat{p} .

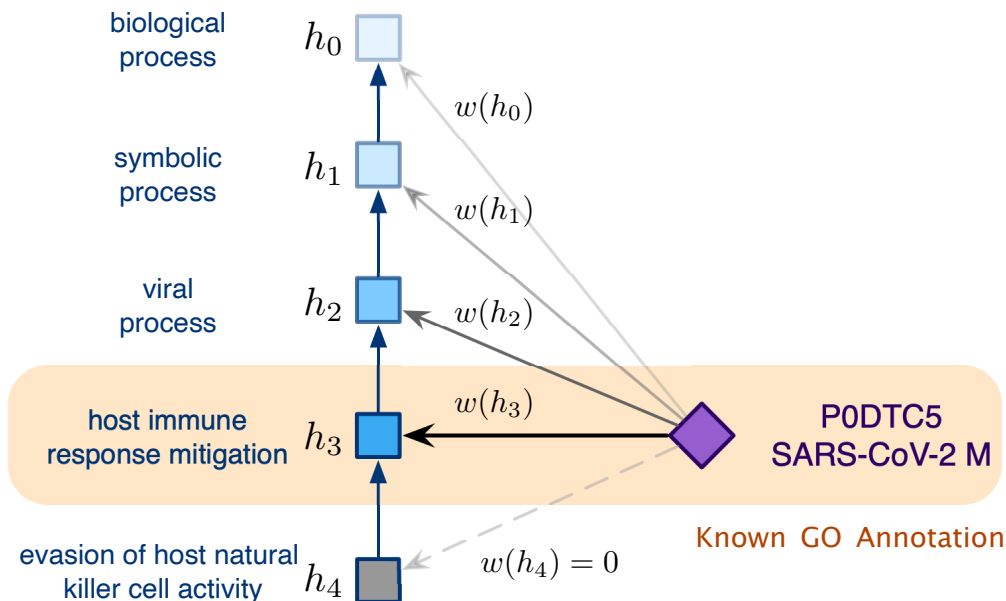


Figure 3.4: Explanation of weighted transfer model for modeling hierarchical gene ontology.

Exemplarily, there could be several ways to calculate the association weight.

Level-based weight. The level of the node in one hierarchical taxonomy is a natural indicator of its specificity. Accordingly, the weight can be defined as,

$$\omega(o) = \frac{l}{l_{\max}}$$

where l is the term's current depth and l_{\max} is the maximum length of the associated branch in the gene ontology DAG.

Degree centrality weight. A small node's degree centrality in the graph roughly reflects

its specialty and we apply

$$\omega(o) = \frac{1}{d(o)}$$

as the balance factor for different GO term specialty.

In practice, incorporating a specificity-based weight to the transfer model essentially enhances the inference in the protein domain, as we have observed in the evaluation in Section 3.4. However, the above weight options generally yield similar performance gain, and we fix the weight option as the level-based weight in our experimental setting.

3.3.4 Joint Learning Objectives

Bio-JOIE jointly learns two knowledge models respectively for GO term relations and PPIs, and a transfer model to support knowledge transfer between these two. Therefore, the joint learning objective minimizes the following loss,

$$\mathcal{L} = \lambda^t \mathcal{L}_T + \lambda^p \mathcal{L}_K^{\mathcal{G}_p} + \mathcal{L}_K^{\mathcal{G}_o}$$

λ^p and λ^t are two positive hyperparameters. We use Adam [KB15a] to optimize the learning objective loss. The learning process uses orthogonal initialization [SMG14] to initialize the weight matrix, and Xavier normal initialization [GB10] for vector parameters. A normalization constraint is enforced to keep all embedding vectors of GO terms and proteins on unit hyper-spherical surfaces, which is to prevent the non-convex optimization process from collapsing to a trivial solution where all vectors shrink to zero [BUG13, MCW18, YYH15, HCY19].

Note that Bio-JOIE is suitable for joint representation learning on proteomic knowledge of different species. In this protein-GO example, the proteins of these species are significantly different from each other. However, they share the same set of annotations in the GO domain. Therefore, More specifically, if we have multiple PPI networks $\mathcal{G}_i, i = 1, 2, \dots, m$ where m denotes the number of independent species, n knowledge models are trained respectively. Consequently, one unique transfer model is also trained to facilitate the protein-GO knowl-

edge transfer regarding each species. The learning objective on the multi-species setting is changed accordingly as,

$$\mathcal{L} = \sum_{i=1}^m \lambda_i^t \mathcal{L}_T + \sum_{i=1}^m \lambda_i^p \mathcal{L}_K^{\mathcal{G}_p} + \mathcal{L}_K^{\mathcal{G}_o}$$

with the assumption that the knowledge model for gene ontology remains unchanged.

In addition to joint learning on multiple species, Bio-JOIE can also be re-trained from new observations of PPIs. For example, suppose newly discovered SARS-CoV-2-Human PPI knowledge extends the original human PPI networks, we can fine-tune the Bio-JOIE from the saved model and obtained embeddings, by only optimizing the Bio-JOIE on the new triples and hence fast obtain representations for all new proteins, without a long time for retraining the Bio-JOIE from scratch.

3.4 Results

In this section, we evaluate the embeddings learned from Bio-JOIE with two groups of tasks: PPI type prediction (Section 3.4.3) and protein clustering based on enzymatic functions (Section 3.4.4). Furthermore, we provide an extensive case study in Section 3.4.5 on SARS-CoV-2 related PPI prediction and classification.

3.4.1 Dataset

The protein-protein interactions for three species, yeast (*Saccharomyces cerevisiae*), fly (*Drosophila melanogaster*), and human (*Homo sapiens*) are collected from STRING [SMC16] database. There are seven types of interactions annotated in the STRING database. To preserve a balanced and sufficient number of cases in each class, we randomly choose the protein pairs from four types of interaction: activation, binding, catalysis, and reaction. In total, there are 21704, 10000, 36400 pairs of proteins for yeast, fly, and human, respectively; each type contains roughly the same number of interactions. Table 3.1 summarizes the PPI information for each species. Note that, the human PPI dataset does not contain the virus-generated proteins, but the set partially overlaps with the virus-human pan-PPI

networks.

The gene ontology annotations for each protein are extracted from gene ontology Consortium [Con18], including all three biological aspects: biological process (BP), cellular components (CC), and molecular function (MF). Table 3.2 summarizes the number of relations between proteins and GO terms. The relations between GO terms include *is-a*, *part-of*, *has-part*, *regulates*, *positively-regulates*, and *negatively-regulates*.

Table 3.1: Statistics of PPI networks and associated GO annotations from different species.

Species	# Proteins	# PPI Triples	# GO Annotations
Yeast	3,736	21,704	191,801
Fly	3,826	10,000	87,807
Human	8,204	36,400	102,759

Table 3.2: Statistics of three aspects in the gene ontology: biological processes (BP), cellular components (CC) and molecular functions (MF).

Aspects	BP	CC	MF
# GO entities	5744	1,147	1,764
# GO triples	19,021	2,116	2,190
# Protein-GO annotations (yeast)	72,956	58,729	60,116
# Protein-GO annotations (fly)	44,605	24,550	18,652
# Protein-GO annotations (human)	42,899	32,929	26,931

For the SARS-CoV-2 dataset, we collect the latest virus-protein interaction from BioGrid¹ and the limited GO annotations for SARS-CoV-2 from Gene Ontology Consortium², as last updated on early April. In summary, there are 26 SARS-CoV-2 generated proteins and 332 human proteins presenting evidence of viral-human protein interactions as suggested by (author?) [GJB20]. The selection is based on a high MIST score and a low SAINTexpress BFDR from Affinity Capture-MS. Out of the same experiment, we select 1131 viral-human protein pairs with MIST scores lower than 0.01 as our negative samples. The 26 SARS-CoV-2 generated proteins are annotated with 282 GO terms. In addition to SARS-CoV-2, BioGrid also includes 30 viral proteins from SARS-CoV and MERS-CoV, which are two similar

¹Data source: <https://wiki.thebiogrid.org/doku.php/covid>

²Data source: <http://geneontology.org/covid-19.html>

contagious viruses causing respiratory infection. These 30 viral proteins are annotated with 630 GO terms, and display 326 interactions with human proteins. All processed datasets are available at <https://www.haojunheng.com/project/goterm>.

3.4.2 Baselines

We compare our model Bio-J0IE with Onto2Vec [SGH18] the most applicable state-of-the-art approach, on learning the representation of proteins. Onto2Vec considered the annotation from gene ontology for representation learning. In addition, we compare Bio-J0IE with a simpler setting, Bio-J0IE-NonGO, where we only consider the single-domain knowledge of PPI.

Onto2Vec, Onto2Vec-Parent, Onto2Vec-Ancestor. Onto2Vec utilizes the annotation information from gene ontology to create pairwise context and apply Word2Vec [MSC13] to generate protein and GO term embeddings. Its schema allows the model to learn the representation of proteins and GO terms simultaneously. The proposed setting of Onto2Vec only includes the direct relationship between a protein and a GO term. In this experiment, we explicitly include the relationship between a protein and the parents of the annotated GO terms, named *Onto2Vec-Parent*, and the ancestors of the annotated GO terms, named *Onto2Vec-Ancestor*.

Onto2Vec-Sum, Onto2Vec-Mean. To examine the effect of Onto2Vec on learning the protein representation from a single domain, i.e. gene ontology, we remove the relations between proteins and GO terms during the learning process. The representation of a protein is then computed by either summing up the embeddings of all the associated GO terms (*Onto2Vec-Sum*), or taking the average of the embeddings of those GO terms (*Onto2Vec-Mean*).

OPA2Vec Based on Onto2Vec, OPA2Vec further learns the protein and GO term embeddings by leveraging meta-data (labels, synonyms, etc), which better characterize GO terms.

Bio-J0IE (NonGO). As opposed to considering the knowledge from a single domain of

gene ontology, we adopt Bio-JOIE to consider only the knowledge from Protein-Protein Interaction. In this approach, all the gene ontology annotations and the gene ontology graph are neglected, and thus is reduced to a knowledge model. We only use the knowledge model in Section 3.3.2, where the protein embeddings are solely learned from PPI networks by the original KG embedding technique, DistMult. We refer to this approach as “Non-GO”.

It is worth mentioning that the goal of Onto2Vec and OPA2Vec is to learn the protein representation; therefore, to adapt for the task of PPI prediction, we concatenate the embeddings of each pair of proteins and train a multi-class classifier to predict the PPI type for a given pair of query proteins. We examine the performance with four different classifiers: logistic regression (LR), support vector machine (SVM), random forest (RF), and neural networks (MLP). The evaluation is conducted with five-fold cross-validation. Similar settings apply to all Onto2Vec variants and OPA2Vec. On the contrary, our proposed model equips with relational modeling and outputs PPI predictions by selecting the most plausible relation type. As a result, we do not need an additional classifier for Bio-JOIE and Bio-JOIE-NonGO.

3.4.3 PPI Type Prediction on Multiple Species

We examine how effectively Bio-JOIE leverages gene ontology to predict protein-protein interaction types. To do so, we first evaluate the performance on three organisms separately: human, yeast, and fly. Then we study the contribution of the three aspects in gene ontology, i.e. biological process (BP), cellular component (CC), and molecular function (MF), on predicting the type of PPI. Specifically, we provide an analysis on how the knowledge from Gene Ontology contributes to PPIs in different species.

Experimental setting. We first separate the PPI triples into approximately 70% for training, 10% for validation and 20% for testing. For hyperparameters with the best performance from the validation set, we select dimension $d_p = d_o = 300$ and margin parameters $\gamma^G = 0.25$, $\gamma^A = 1.0$ and $\gamma_{HA} = 1.0$. Two weight factors in the joint learning objective are set as $\lambda_p = 1.0$, $\lambda_t = 1.0$. We use DistMult for the knowledge model in Section 3.3.2,

with hierarchy-aware regularization and the level-weighted transfer model (Section 3.3.3) deployed. For simplicity, the reported Bio-JOIE adopts the same settings if not specifically explained. The number of epochs in training on all settings is limited to 150. For evaluation, we aim at predicting the correct interaction type, given pairs of proteins in the test set. We conduct a 5-fold cross validation for Bio-JOIE and all baselines, and report the average and standard deviation of accuracy. The best-performing classifier is RF for OPA2Vec and most of the Onto2Vec variants. The only exception is to apply MLP for Onto2Vec-Ancestor on fly.

Table 3.3: PPI type prediction accuracy (%) evaluated on yeast, fly and human species.

Model	Yeast	Fly	Human
Onto2Vec	76.41 ± 0.73	70.85 ± 0.85	77.97 ± 0.46
Onto2Vec-Parent	80.79 ± 0.66	75.46 ± 1.11	74.90 ± 0.46
Onto2Vec-Ancestor	86.31 ± 0.42	80.31 ± 0.92	78.73 ± 0.46
Onto2Vec-Sum	76.38 ± 0.83	72.84 ± 1.13	72.53 ± 0.73
Onto2Vec-Mean	77.95 ± 0.81	74.38 ± 1.13	73.47 ± 0.80
OPA2Vec	79.88 ± 0.74	74.45 ± 0.97	72.04 ± 0.58
Bio-JOIE-NonGO	83.65 ± 0.92	77.58 ± 1.07	76.10 ± 0.87
Bio-JOIE	87.15 ± 1.15	84.56 ± 0.81	81.42 ± 0.62
Bio-JOIE-Weighted	90.12 ± 1.21	85.55 ± 1.57	83.89 ± 0.92

Results. The results for PPI type prediction are shown in Table 3.3. We observe that our best Bio-JOIE variant outperforms Bio-JOIE-NonGO by 7.4% on average for all three species. This observation directly shows that gene ontology KG provides complementary knowledge for proteins. Subsequently, Gene Ontology annotations benefit the learning of protein representations and better predict the interaction types between proteins. Compared to other baselines, it is observed that Bio-JOIE notably outperforms Onto2Vec-Ancestor with an average increase of 7.4% on the prediction accuracy, and a relative gain of 9.0% on average of all three species. This observation is due to the advantage that Bio-JOIE better leverages the complementary knowledge from PPI to enhance the PPI prediction. As mentioned in Section 3.4.2, Onto2Vec does not utilize the PPI information into protein embedding learning. Instead, it obtains embeddings based on the aggregated semantic representations of GO terms. It requires additional classifiers for PPI type prediction given

pre-trained protein embeddings. In contrast, **Bio-JOIE** jointly learns protein representations from both the knowledge model that captures the structured information of known PPIs, and the transfer model that delivers the annotations of GO terms. Also, we observe that **Bio-JOIE-Weighted** achieves better results than **Bio-JOIE**, with a relative performance gain of 2.5%. We hypothesize that such gain is attributed to specificity modeling in the transfer model which distinguishes more specific and informative GO terms from other general GO terms and assigns a higher weight, which selectively learns the alignments between two domains. In terms of different species, we also observe that **Bio-JOIE** achieves a higher PPI prediction accuracy on yeast compared to human and fly. The possible reason is that the yeast interaction network is denser, such that 0.30% of the protein pairs are known to interact, compared to human (0.13%) and fly (0.11%), which indicates that yeast is possibly well studied. **OPA2Vec** claims to be an improved version of **Onto2Vec**. Similar to **Onto2Vec**, it only considers the direct relationship between a protein and a GO term, without parents and ancestors. We find that **OPA2Vec** performs slightly better than **Onto2Vec** on Yeast and Fly, but worse on Human. In addition, **OPA2Vec** falls short when compared to any of the **Bio-JOIE** variants, indicating that incorporating the metadata of GO terms is insufficient for protein representation learning.

It is noteworthy that unlike **Onto2Vec**, which achieves its best performance with the help of full gene ontology (i.e. **Onto2Vec-Ancessor**), our **Bio-JOIE** model can utilize only the GO terms that are directly annotated with the proteins to accomplish the highest accuracy score. This also makes **Bio-JOIE** training processes more time efficient. We hypothesize that for **Bio-JOIE** in the PPI type prediction task, GO terms that are directly related to associated proteins with high specificity are sufficient for the transfer model to model the protein-GO association in the embedding spaces. In contrast, **Onto2Vec** needs entire structured information of GO terms for its word2vec module to construct an exhaustive context of protein features.

We further explore the effects of three different aspects of gene ontology in predicting the types of PPIs. To achieve this, we train **Bio-JOIE** in settings where only specific aspects of

Table 3.4: Comparison of PPI prediction accuracy of Bio-JOIE on three different aspects of gene ontology.

#	Aspects	Yeast	Fly	Human
1	BP	0.8794	0.8402	0.8153
	CC	0.8499	0.8272	0.8054
	MF	0.8539	0.8386	0.8165
2	BP+CC	0.8717	0.8473	0.8271
	BP+MF	0.8673	0.8471	0.8163
	CC+MF	0.8569	0.8466	0.8170
3	AllGO	0.9012	0.8555	0.8389

gene ontology annotations are used. Results are shown in Table 3.4, in which BP, CC and MF respectively refer to the cases where GO terms of *biological processes*, *cellular components* and *molecular functions* are used. “BP + CC” denotes that the GO terms from both biological processes and cellular components are included in training. We observe that Bio-JOIE performs the best with GO terms from all aspects (full gene ontology). This phenomenon is consistent among all three species, indicating that the protein representations are more robust when learning from a more enhanced knowledge graph. It is also interesting to see that the accuracy of the task is generally higher when we include the GO terms from biological processes. This leads to 2.61% improvement in accuracy over CC, and at least 2.13% of improvement over MF when evaluated individually. In the two-aspect evaluation, “BP+CC” is in average leads to 0.7% better accuracy than “CC+MF”. This is attributed to the fact that BP is the largest group in the gene ontology, containing more entities and relational facts. Consequently, Bio-JOIE achieves the best performance with all three aspects of gene ontology annotations incorporated. This indicates that the characterization of PPIs benefits from more comprehensive gene ontology annotations.

Table 3.5: PPI type prediction accuracy on different configurations of multi-species joint learning.

Model	Yeast	Fly	Human
Bio-JOIE (single)	0.9012	0.8555	0.8389
Bio-JOIE (concat)	0.8795	0.8282	0.8028
Bio-JOIE (multi-way)	0.9062	0.8638	0.8426

In addition to joint learning from two different domains (i.e. GO terms and PPIs), as mentioned in Section 3.3.4, Bio-JOIE can be trained to capture PPIs for multiple species with several species-specific knowledge models, along with transfer models that bridge the universal gene ontology. To validate the benefit of joint learning on multiple species together, we consider three following configurations of Bio-JOIE: (i) the “multi-way” setting uses one unique knowledge model and one transfer model to the universal gene ontology for each species; (ii) the “concat” setting uses one unified knowledge model to capture all species of PPIs, together with one transfer model to learn protein-GO alignments, that is, simply concatenate all PPI triples and all gene ontology annotations of proteins in multiple species; (iii) the “single” setting trains separately on each species, which is exactly the same as in the setting in Table 3.3. We summarize the results in Table 3.5. It is observed that the “multi-way” setting can slightly improve PPI performance in comparison to the “single” setting that trains separately on each species. Also in the “concat” setting with one shared transfer model and knowledge model, the performance significantly drops with a 2.8% decrease of accuracy on average compared to the “single” setting. Such results suggest that each species has unique patterns of PPIs, such differences are better differentiated in separate embedding spaces. Hence, the multi-way setting better encodes the species-specific knowledge and model, which helps the type prediction of PPIs for each species by Bio-JOIE that are jointly trained on multiple species.

3.4.4 Identifying Protein Families And Enzyme Commission Based Clustering

Besides inferring PPI types, the embedding representations of proteins can also be used to identify potential protein families based on their functions. This can be achieved by performing clustering algorithms on the learned protein embeddings.

The Enzyme Commission number (EC number) defines a hierarchical classification scheme that provides the enzyme nomenclature based on enzyme-catalyzed reactions. The top-level EC numbers contain seven classes: oxidoreductases, transferases, hydrolases, lyases, isomerases, ligases, and translocases. In this experiment, we select 1340 yeast proteins in total

with enzymatic functions. We learn the protein representations using all the triples of PPI networks and the annotation from gene ontology and evaluate the learned representations of these proteins by performing the k-means clustering algorithm to group them into seven non-overlapping clusters. These clusters are compared with the top-level of enzyme commission classification. Purity score is reported as evaluation metrics.

The evaluation of the clustering results is shown in Table 3.6. Bio-JOIE achieves the best clustering performance on yeast by a relative increase of 9.7%, which demonstrates that Bio-JOIE has the good model capability to representation learning and empirically show the validity of the learned embeddings to measure the similarity. We hypothesize that Bio-JOIE better incorporates protein annotation resource and utilizes the complementary knowledge in the gene ontology domain, while Bio-JOIE also captures PPI information and encode it into protein embeddings. This in the end results in comprehensive representations for proteins and helps to identify protein EC classes by clustering.

Table 3.6: Results of top-level EC clustering by K-means on learning selected yeast protein embeddings.

Model	Purity Score
Onto2Vec	0.2339
Onto2Vec-Parent	0.2452
Onto2Vec-Ancestor	0.3224
Onto2Vec-Sum	0.3022
Onto2Vec-Mean	0.2616
Bio-JOIE (KM only)	0.2514
Bio-JOIE	0.3306

3.4.5 Case Study: SARS-CoV-2-Human Protein Target Prediction

The COVID-19 pandemic requires much effort and attention from scientists in different fields. However, there is very limited knowledge of the molecular details of SARS-CoV-2. In this subsection, we apply Bio-JOIE to gain more insights into the PPI network between SARS-CoV-2 and human proteins. Specifically, we explore the potential of Bio-JOIE in predicting whether a pair of human and SARS-CoV-2 proteins interact or not. This is modeled as a

binary prediction task. Correspondingly, results from the binary predictions can serve as a guide to identify the targeted proteins by SARS-CoV-2. We first use the known interactions between these two species to validate the effectiveness of Bio-JOIE. These interactions are experimentally verified as described in Section 3.4.1. In this setting, we particularly study the contribution of the knowledge of other closely related viruses (SARS-CoV and MERS) on supporting PPI prediction. We also show the high-confidence candidates of targeted human proteins predicted by Bio-JOIE for four selected SARS-CoV-2 proteins.

Experimental setting. In this experiment, we randomly split the known positive human-virus PPIs into train and test sets with a ratio of 80% and 20%. We train Bio-JOIE on this train set along with human PPIs. For evaluation, positive test samples and selected negative samples, mentioned in Section 3.4.1 are used to perform binary prediction. We adopt F1-score as the evaluation metric.

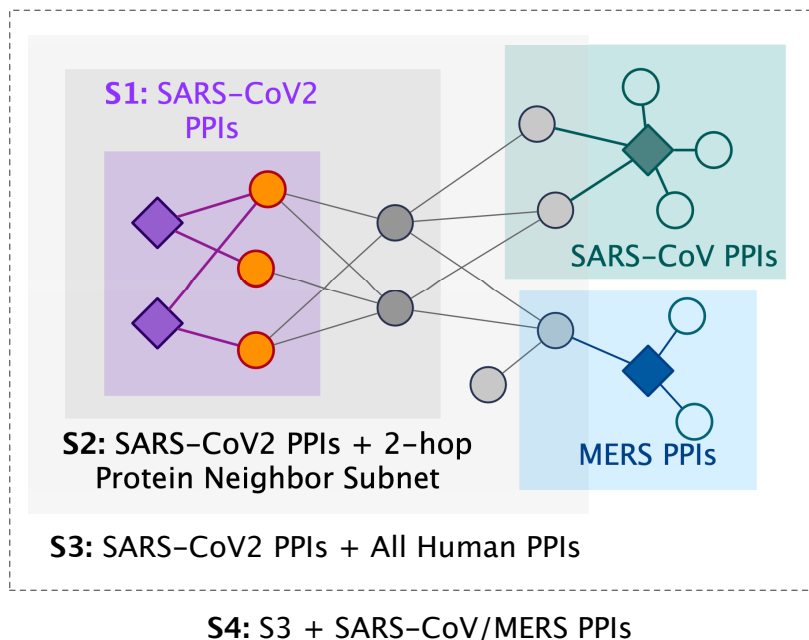


Figure 3.5: Different scopes of input to train Bio-JOIE for SARS-CoV-2 PPI prediction.

Results. As in Section 3.4.3, we first evaluate Bio-JOIE on SARS-CoV-2 PPI prediction. From the observation in Section 3.4.3, two important factors are considered: three aspects in the gene ontology domain and the scope of input SARS-CoV-2-Human PPIs. More specifically, we define increasingly four scopes of input PPIs, as shown in Figure 3.5, i.e. (1) S1:

Table 3.7: F-1 score on SARS-CoV-2-Human PPI interaction classification.

Input	S1	S2	S3	S4
NonGO	0.6737	0.7004	0.6918	0.6997
BP	0.7103	0.7353	0.7348	0.7492
CC	0.7188	0.7383	0.7380	0.7675
MF	0.6737	0.7016	0.7022	0.7365
BP+CC	0.7257	0.7570	0.7499	0.7813
BP+MF	0.7252	0.7479	0.7486	0.7713
CC+MF	0.7317	0.7622	0.7692	0.7917
AllGO	0.7307	0.7537	0.7500	0.7885

Only using the train folds of SARS-CoV-2-Human PPIs; (2) S2: Using SARS-CoV-2-Human PPIs with the 2-hop neighbor proteins from SARS-CoV-2 viral proteins, i.e. including the ones which also interact with any proteins that the SARS-CoV-2 interacts; (3) S3: SARS-CoV-2-Human PPIs with all other protein interactions on human; (4) S4: SARS-CoV-2-Human PPIs with all protein interactions in S3 plus all SARS-CoV and MERS PPIs. As for the aspects of the gene ontology domain, similar to Table 3.4 in Section 3.4.3, we adopt eight options, i.e. one without gene ontology information (NonGO), three using a single aspect of GO terms (BF, CC, MF), three options using two of the aspects (BF+CC, etc) and one using all three aspects (AllGO).

The results are summarized in Table 3.7. In terms of gene ontology aspects, we observe that CC contributes the most compared to other aspects of gene ontology annotations, and the best performance is achieved by adopting CC+MF in Bio-JOIE learning. One explanation is that most of the SARS-CoV-2 proteins have CC annotations and these annotations make up over 70% of all currently available annotations on average. However, less than 5 proteins (such as NSP and ORF 1a) have BF and MF annotations, possibly due to insufficient knowledge on understanding SARS-CoV-2 biological mechanism. As for the input fields, we find that the performance drastically increases with the expansion of input from S1 to S2, which indicates that interactions of 2-hop neighbor proteins can benefit SARS-CoV-2 PPI prediction. However, such a trend is not clearly observed when expanding the input field from S2 to S3. We hypothesize that proteins that are not within 2-hop neighbors may not

be very related to SARS-CoV-2 or provide beneficial insights. Interestingly, when adding interactions of two related coronaviruses (SARS-CoV/MERS-CoV) that cause respiratory infection, the performance continues to improve with a relative gain of 3.4%. As shown in Figure. 3.2, viruses that are closely related to SARS-CoV-2 tend to share important properties. This strongly suggests that it is crucial to leverage their interactions and gene ontology annotations as augmented knowledge for drastically emerging SARS-CoV-2.

Table 3.8: Top target proteins predicted by Bio-JOIE. Known interactions from training set are excluded. Proteins that are considered as high-confidence targets are boldfaced.

SARS-CoV-2	Targeted proteins in human
ORF8	P05556 , P61019, Q9Y4L1 , P17858, Q92769, Q9BQE3, Q9NQC3, Q9NXK8 , P33527, P61106
NSP13	Q99996 , P67870, P35241 , O60885, P26358, Q9UHD2 , Q12923, Q86YT6, Q04726 , P61106
M	P26358, Q9NR30, O75439 , Q15056, P61962, P49593, P33993, O60885, Q9Y312 , P78527
NSP7	P62834, P51148 , P62070, P67870, O14578, Q8WTV0 , P53618, Q9BS26, O94973, Q7Z7A1

Besides providing PPI prediction, the proposed model can help by identifying high-confidence candidates for potential human protein targets; this is considered as a link prediction task. When a viral protein (such as SARS-CoV-2 M protein) is given as the query, along with a specific relation (such as “binding” under the experiment system type of “Affinity Capture-MS”), Bio-JOIE can output a list of most likely protein targets by enumerating the triples with top $f_r(h, t)$ scores. The predictions are listed in Table 3.8. It is our observation, Bio-JOIE can successfully predict the high-confidence human protein targets in the test set from by [GJB20] among its top predictions (marked as boldfaced entities). Other than the proteins in the test set, Bio-JOIE can also provide a list of reasonable candidates that possess a relatively high MIST score. For example, as shown in Figure 3.6, P62834 is one of the top-ranked protein targets of SARS-CoV-2 NSP7 by our Bio-JOIE, which has a MIST score of 0.658. Diving deep into the facts for P62834, though P62834 is not considered a high-confidence target by [GJB20], we observe that both P62834 (RAB1A_HUMAN) and SARS-CoV-2 NSP7 interacts with protein P62820 (RAB1A_HUMAN). Besides, they are

both annotated with the cellular component GO:0016020 (membrane) and enables molecular function GO:0000166 (nucleotide binding), which are possibly the reasons for Bio-JOIE making such a prediction with a high rank. Furthermore, Bio-JOIE’s predictions include proteins that are not covered by [GJB20], which inspires further scientific research to verify.

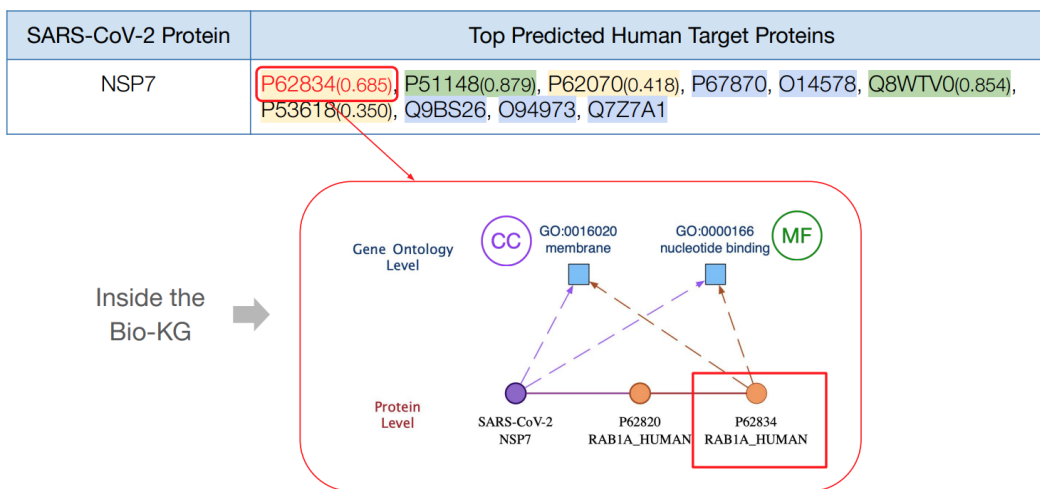


Figure 3.6: Connection paths between SARS-CoV-2 NSP7 (viral protein) and Protein:P62834 in the SARS-CoV-2 PPI knowledge graphs.

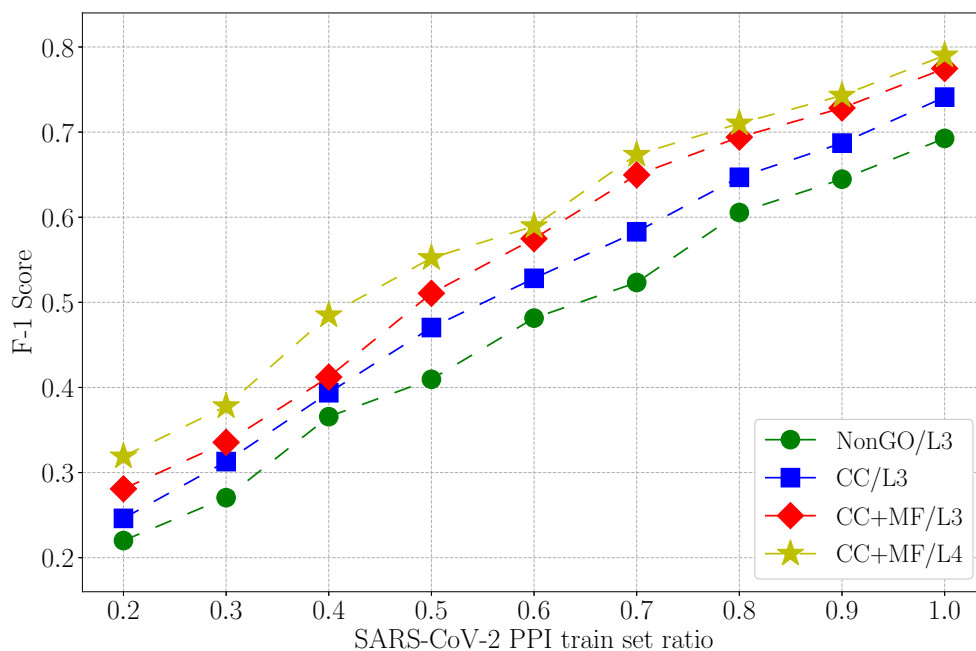


Figure 3.7: Bio-JOIE performance on different train-set ratios of SARS-CoV-2-Human PPIs.

We further investigate how the information sufficiency of SARS-CoV-2 related PPIs in

the training set affect the performance. We define the train-set ratio parameter as means the proportions of the SARS-CoV-2-Human PPIs that are used for training Bio-JOIE and follow the aforementioned evaluation protocol on “NonGO/S3”, “CC/S3”, “CC+MF/S3” and “CC+MF/S4” as input other than the control of SARS-CoV-2-Human PPIs part. We plot the PPI results in Figure 3.7. As expected, when the proportion of SARS-CoV-2-Human PPIs used for training increases from 20% to 80%, the F1 score improves from 0.2-0.3 to around 0.8, which strongly confirms that the known SARS-CoV-2-Human PPIs serve as one significant factor to the PPI prediction. Moreover, the more knowledge we know about existing SARS-CoV-2 interaction, the more powerful the model is to predict SARS-CoV-2. We also observe that the performance is not saturated when the training ratio is approaching 100%, which possibly results from the fact that as a novel coronavirus, the current known interactions are still very limited. This encourages the scientific communities to unearth more knowledge on SARS-CoV-2; moreover, Bio-JOIE has the potential of bringing about significant advances based on new discoveries.

3.5 Extension: Bio-JOIE Inference on Texera (Collaborative Machine Learning Demonstration)

In previous sections, we have introduced one important case study for SARS-CoV-2 viral and human protein interactions and provided inspiring results for potential drug development. This is a good example to combine machine learning and computational bioinformatics, to accelerate the process of drug development. It is also a great demonstration for undergraduate students and junior student researchers to think about the huge potential and long vision of machine learning applications. Therefore, we contribute special outreach efforts to make the model and its entire workflow which uses knowledge graph embedding for drug repurposing more accessible, understandable and collaborative to broader communities and younger generations who are passionate about machine learning.

In this section, with the help of Texera, we make the training and inference process of

Bio-JOIE and SARS-CoV-2 drug repurposing as one collaborative workflow, just like shared Google Document or Slides. Texera is a system to support collaborative, ML-centric data analytics as a cloud-based service using GUI-based workflows. It supports scalable computation with a parallel backend engine, and enables advanced AI/ML techniques [WKN20]. The motivation is that, most of the existing ML applications are often managed with scripts to run training and inference jobs. While it is common for ML practitioners who have coding expertise, the scripts with lines of pure codes are no doubt challenging to those who do not have sufficient coding experience, which presents a steep learning curve for non-coding collaborators and take huge efforts in communications (such as how to change the code lines to make adjustment in ML models) in a typical interdisciplinary collaboration. Though some notebook-style coding interfaces (IDEs) (such as Jupyter Notebooks or Google Colab) provide some clarifications and comments on code with explicit text blocks and interactive execution, such challenges remain to limit the full capacity of collaboration on machine learning applications.

Our proposed transformation is from such scripts and notebooks to executable workflows with accessible modules to every collaborator. The overview of the a general transformation process is shown in Figure 3.8. The entire complex script which contains a series of steps such as model and test data loading, processing and result output, are changed into different and sequentially connected modules, mostly as User Defined Functions (UDFs), which ultimately form a “Lego-style” workflow to execute the same ML jobs.

We transform the Bio-JOIE inference steps utilized by knowledge graph embedding into several components for a better understanding of the entire process. As shown in Figure 3.9, the inference procedure which is considered similar as “link prediction” between SARS-CoV-2 proteins and FDA-approved drugs, following [ISM20]³, consists of self-explaining connected modules *model loading*, *test case input*, *embedding lookup*, *score computation*, *ranker* and *results output and evaluation*.

³https://github.com/gnn4dr/DRKG/blob/master/drug_repurpose/COVID-19_drug_repurposing.ipynb

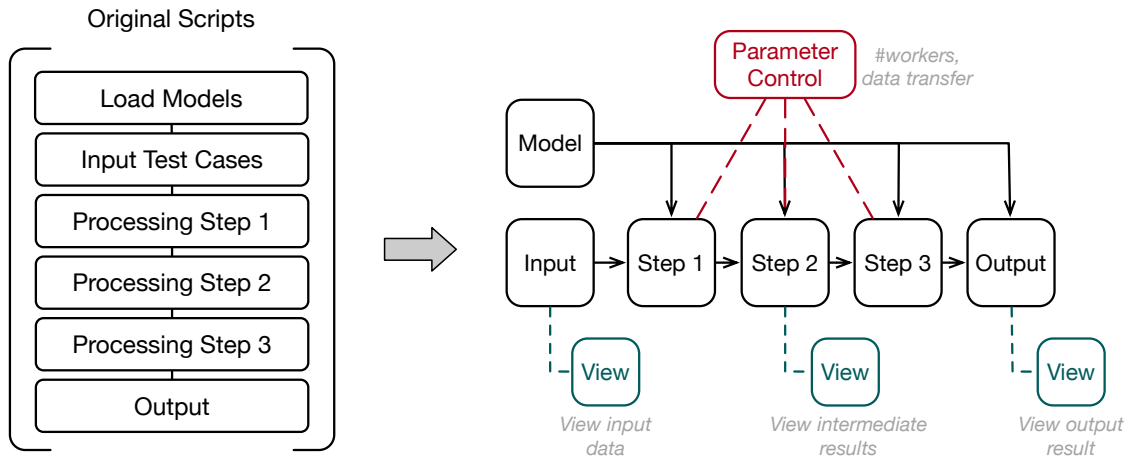


Figure 3.8: A high-level overview of the script to the modularized workflow of ML applications.

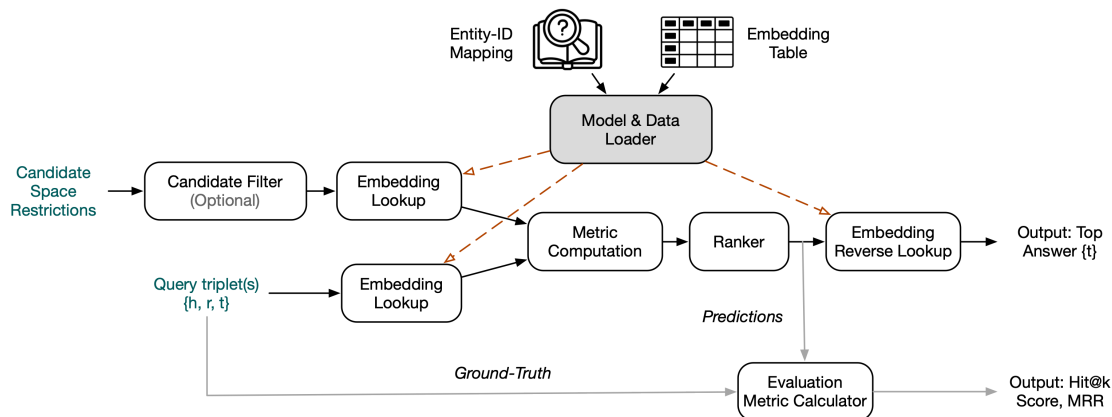


Figure 3.9: Detailed steps of original knowledge graph triple completion problem. While all steps can be included in one single script, it can be decomposed into multiple modules that are easy to understand and control.

With deployment on Texera, the workflow successfully runs and produces results for SARS-CoV-2 drug repurposing, as shown in the screenshot (Figure 3.10), which provides interactively and collaborative inferences to visualize and analyze model results.

3.6 Conclusion

In this chapter, we present a novel model Bio-JOIE, that enables end-to-end representation learning for cross-domain biological knowledge bases. Our approach utilizes the knowl-

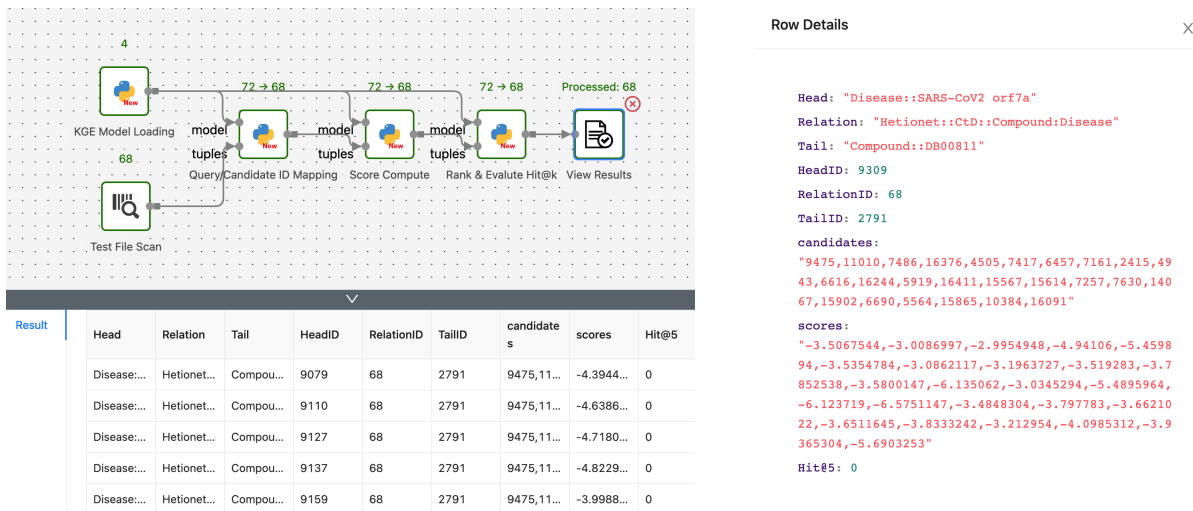


Figure 3.10: Inference steps on SARS-CoV-2 drug repurposing are successfully running on Texera. Each row represents one single query such as {Disease:SARS-CoV-2 ORF7a, Relation: Disease-Compound, Compound:?}. With one click, the entire information of the corresponding row will be presented in detail.

edge model to capture structural and relational facts within each domain and motivates the knowledge transfer by alignments among domains. Extensive experiments on the tasks of PPI type prediction and clustering demonstrate that Bio-JOIE can successfully leverage complementary knowledge from one domain to another and therefore enable learning entity representation in multiple interrelated and transferable domains in biology. More importantly, Bio-JOIE also provides interaction type predictions on SARS-CoV-2 with human protein targets, which potentially brings reliable computational methods seeking new directions on drug design and disease mitigation.

In our main directions of future research, we plan to enhance and extend entity representations by systematically incorporating important multimodal features and annotations. For example, primary sequence information and secondary geometric folding features can be modeled simultaneously in protein networks and their combined representation can lead to a comprehensive understanding that will greatly benefit many downstream applications.

With the development of Bio-JOIE, we realize that compared to the complex biological and biomedical knowledge graphs, a two-view formulation of gene ontology and protein is still one simplified version of understanding the interactions between the biological entities

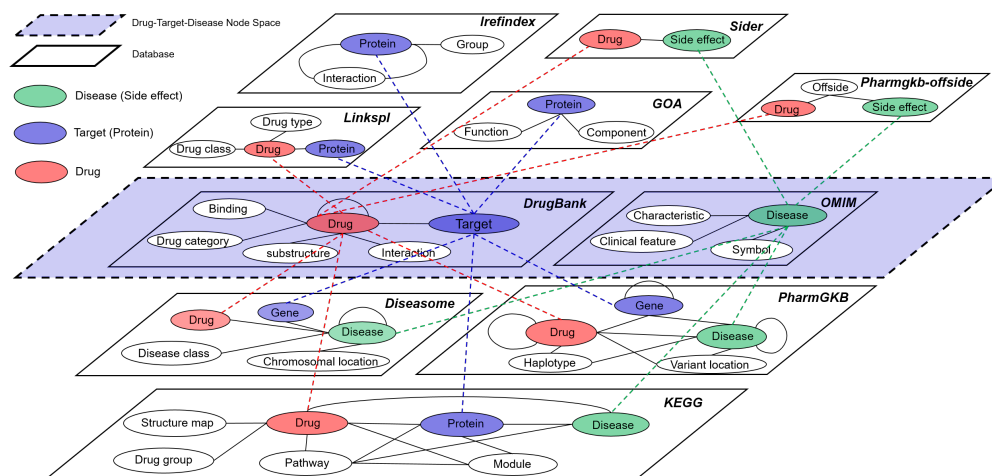


Figure 3.11: Biological knowledge graphs for drug-target discovery from BETA benchmark [ZLW22].

and cannot fully cover the information from other related domains, like human genetics, human tissue profiling, function genomics and clinical record [GDJ21]. One recent work for drug discovery benchmark [ZLW22] has investigated a comprehensive study on more types of biological entities including diseases, drugs, proteins, and side effects, based on multiple resources and alignments, as shown in Figure 3.11. One of the promising future directions is that, with more information on biological KGs as a backend to better understand the interactions within metabolic reactions, not limited to gene ontology and proteins, more accurate and reliable predictions can be made on emerging diseases, as a new milestone for computational bioinformatics by further advancing research of natural sciences and a huge benefit public health and social welfare.

CHAPTER 4

P-Companion: A Product Graph Based Principled Framework for Diversified Complementary Product Recommendation

4.1 Introduction

If one customer “Pablo” wants to buy a tennis racket, what are the best 3 complementary products to recommend to purchase together? 3 tennis ball packs, 3 headbands, 3 overgrips, or 1 of each respectively? Product complementary recommendation has become increasingly critical for the success of online websites, especially for e-commerce sites such as Amazon, eBay, Taobao, etc. Such recommendations often help customers find a high-quality selection of product complements to purchase together and meet their needs, which is key to enhancing user experience and satisfaction and has strong business value. Figure 4.2 shows one toy example of complementary recommendation in the real-world shopping experience. Given that the customer (possibly a beginner tennis player) shows his intent to purchase a tennis racket (already in the shopping cart, considered as a “query product”¹), it is not satisfactory to recommend three other similar tennis rackets to purchase together in List 1 (considered as “substitutes”). Instead, some complementary items to tennis rackets are expected, such as tennis balls. Moreover, it seems better to have List 3 with one tennis ball pack, one racket cover and one headband respectively as recommendations than List 2 with three packs of

¹We specifically define “query product” in this chapter as the product that serves as recommendation condition, different from the query in search engine. Typically a “query product” refers to the items that customers plan to purchase or put into the cart in the e-commerce scenario.

tennis balls, which indicates that both relevance and diversity are needed to fulfill customer's need.

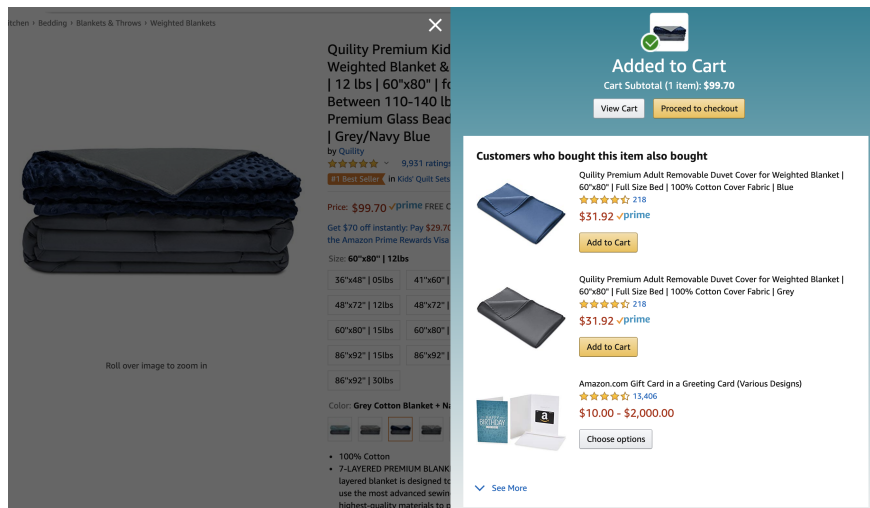


Figure 4.1: One application of complementary product recommendation in Amazon. Multiple complementary items are listed after one item has been put into the cart by the customer.

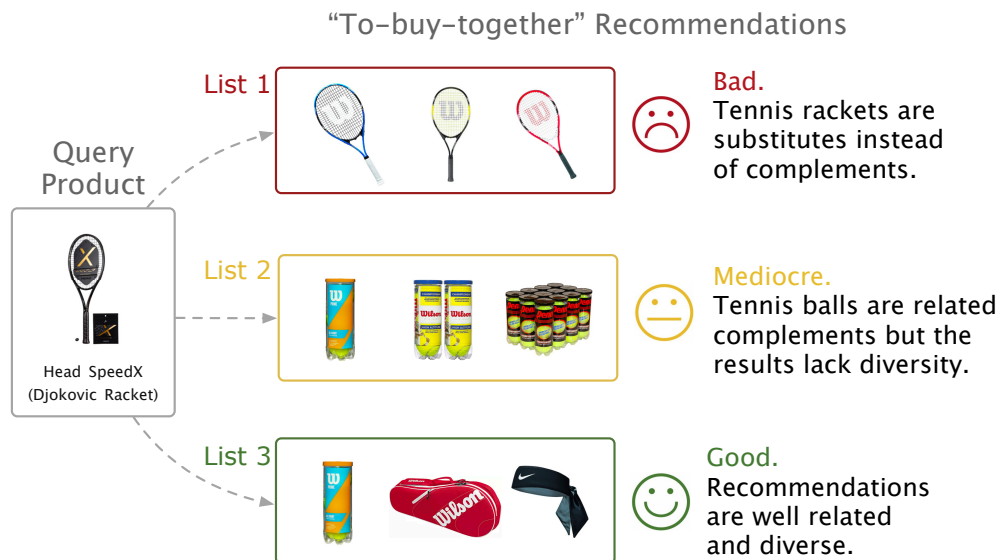


Figure 4.2: Three lists of “to-buy-together” recommendation on the e-commerce platform. Good complementary recommendations require both relatedness and diversity.

Solving such complementary recommendation problem is non-trivial and challenging. From the example in Figure 4.2, it poses at least five particular challenges.

- *C1: Complementary recommendation is not simply based on similarity measurement.*

In many cases, complementary products are not necessary to have similar textual or image features, e.g. tennis rackets and balls. In other words, complements indicate neither high similarity nor low similarity. For example, tennis rackets and balls are not similar to each other on textual or image features.

- *C2: Complementary relationship is not symmetric.* For example, an SD card can be a complement product to a camera but not vice versa, which rules out most of the similarity-based approaches and therefore requires a different mechanism to model such a relationship.
- *C3: Complementary recommendation should be at “product level” in terms of granularity.* The model should be able to make recommendations specifically for one query product by extracting its features.
- *C4: Complementary recommendation suffers from cold-start items.* It is challenging to make recommendations on low-resource items that lack product features or customer behavior data . This also results in low recommendation coverage among all items.
- *C5: Complementary recommendation needs to consider diversity.* The recommendations are typically a set of items with diverse categories and functionalities that provide high utilization for customers’ demand. As shown in Figure 4.2, a diversified recommendation basket, which includes three types of tennis-related products (List 3) is superior to that with only one type (List 2).

While most of the existing methods in recommender systems [SK09] focus on analyzing patterns by frequent pattern mining [HCX07], matrix factorization [KBV09], collaborative filtering [KB15b, SKK01], or other neural network based recommenders [BK16], only a few [HCX07, BBS17, HPM16, LSY03] target at explicitly modeling item-item (product) relationship. Among them, complementary relationship modeling has been scarcely investigated compared to the efforts made for modeling substitutes with similarity-based approaches. Recently, new approaches with behavior-based product graphs, which generally integrate product features and pair-wise relations obtained from customer behavioral data (such as

co-purchase data), have been spotlighted and shown effective on complementary recommendation. Representative examples are Sceptre [MPL15], which proposes a topic modeling to infer networks of products, and PMSC [WJR18], which incorporates path constraints in item-item multi-relational modeling. However, these methods seek to distinguish substitutes and complements, fail to address these aforementioned challenges and dive deep into modeling such properties of complementary recommendation, especially from the diversity perspective. A comparison between P-Companion and existing representative models regarding these challenges are listed in Table 4.1.

Table 4.1: Comparison between P-Companion and existing representative models: Co-Purchase, Sceptre and PMSC.

Property	Co-Purchase	Sceptre [MPL15]	PMSC [WJR18]	P-Companion
Asymmetric (C1, C2)	✓	✗	✓	✓
Product-aware (C3)	✓	✓	✓	✓
Coverage (C4)	✗	✓	✓	✓
Cold-start (C4)	✗	limited	limited	advanced
Diversity (C5)	✗	✗	✗	✓

Researchers are particularly interested in co-purchase data as a strong indicator for complements [MPL15, WJR18]. In real-world applications, we have the following observations with a thorough analysis on such co-purchase data: (i) *co-purchase records between two products may not imply a complement relationship*. We find that product pairs in the co-purchase history can be either substitutes, complements, or even non-related. (ii) *The complement relation among products is often observed in multiple categories*. Existing methods are mostly experimenting within one category such as “electronics” or “grocery”; however, it is quite often for one product under the “electronics” category to have potential complements under “home improvement” or “office” category. In Section 4.2, we show that these two observations with detailed analysis, which are not aligned with the assumptions in previous research.

To address the aforementioned challenges, we propose a novel P-Companion for principled complementary product recommendation. The high-level idea of P-Companion is to leverage the behavior based product graph (BPG) to learn item type transition to predict complementary types, and then perform complementary recommendation regarding different

complementary types. We propose a hierarchical multi-task learning framework, which is an end-to-end complementary recommendation pipeline. The proposed **P-Companion** starts with the foundational **Product2vec** embeddings, which is trained by a new GNN-based product representation learning framework from the product graph, serving as foundation and input to handle large numbers of products. As for recommendation, **P-Companion** first employ an type transition module to predict the related complementary types and then based on the query item together with such types, an type-item projection module is learned to facilitate highly related and diverse complementary recommendation. In summary, our contributions are listed in the following aspects:

- **Data Understanding.** We drop the inaccurate assumptions in existing research on CPR. Based on observations and crowd-sourced annotations on co-purchase and co-view, we propose a new approach to collect labels as distant supervision for CPR.
- **Methodology:** We propose a new model **P-Companion**, that considers both relevance and diversity in CPR modeling and yield diversified recommendations. We also introduce a graph attention based product embedding learning module that makes **P-Companion** robust to deal with cold-start products.
- **Performance:** Through new label collection schema and human evaluation by MTurk, experiments on real-world datasets show that **P-Companion** significantly outperforms state-of-the-art baselines by 7.1% improvement on Hit@10 score, and deliver reasonable and explainable recommendations with diversity across multiple product categories at Amazon in production.

The remainder of this chapter is structured as follows. We formally define the diversified complementary recommendation in Section 4.2, together with a detailed definition of Product Graphs and discussion on how we transit from two-view KG in Chapter 2 to Product Graphs. Section 4.3 describes the proposed **P-Companion** model, and provides details regarding the training and optimization process. Experimental results are shown in Section 4.4. We provide

an overview of the existing techniques related to the work proposed in this paper in Section 4.5 and finally, Section 4.6 concludes and points out directions for future work.

4.2 Preliminaries

In this section, we start with the definitions of related terminology in Behavior-based Product Graph (BPG), together with data analysis and observations in BPGs, then we present the problem formulation for diversified product complementary recommendation.

4.2.1 Behavior-based Product Graph (BPG)

Let \mathcal{I} denote the product item set, \mathcal{C}_i denote item i 's catalog features (e.g. product group and title), and $\mathcal{B} \in \mathcal{I} \times \mathcal{I}$ represent product relationships, (e.g., co-purchase \mathcal{B}_{cp} , co-view \mathcal{B}_{cv} and purchase-after-view \mathcal{B}_{pv})² between pairs of items, which are extracted from customers' historical behaviors. In particular, for each item $i \in \mathcal{I}$, we assume there is an item type $w_i \in \mathcal{C}_i$ that represents product i 's functionality, such as `hdmi-dvi-cable` or `over-ear-headphone`. Similarly, each item may also be associated with a general category, such as `electronics`. Such information can be viewed as a Behavior-based Product Graph (BPG) with products as “nodes”, types and other catalog features as “node attributes”, and pairwise item relationships as “edges”. BPG is essentially a heterogeneous attributed information network as there are three types of edges. Figure 4.3 presents a BPG snapshot of one product with multiple catalog features and related items from different relation connections from customer behaviors.

²More specifically, *co-purchase* means customers who purchased item x also purchased item y ; *co-purchase* means customers who viewed item x also viewed item y ; *purchase-after-view* means customers who viewed item x eventually bought item y

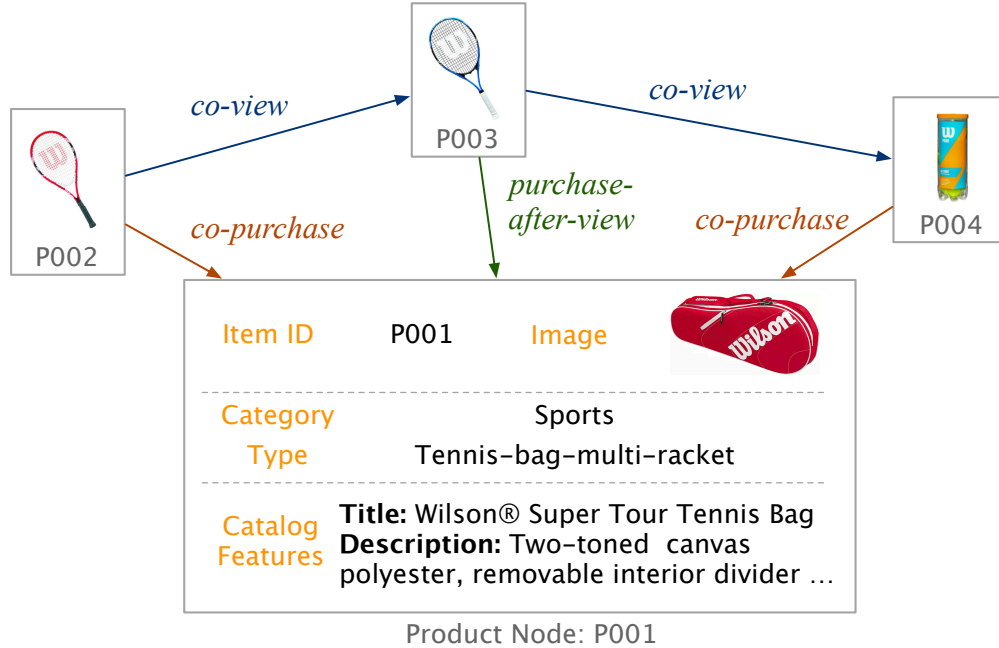


Figure 4.3: One snapshot of a typical BPG. BPG is constructed with nodes as items with catalog features (type, etc) and edges as pairwise relations based on customer behavior.

4.2.2 From two-view KG to two-view PKG

In this section, we briefly discuss the connections between the two-view KG in Chapter 2 and BPG in this work. The most important feature that are shared by KGs and BPGs is that we can observe similar internal structures especially type associations, i.e. "entity-concept" in two-view KG and "product-type" in BPG, as shown in Figure 4.4. Such similarity enable us to model different categories of products and provide a valid way for diversified recommendation modeling, which will be explained in Section 4.3. For completeness, we also include some minor differences between the general KG and the BPG in Table 4.2.

4.2.3 Data analysis in BPG

Section 4.1 mentions our observations, which are not aligned with assumptions and settings in previous research. We conduct a brief data analysis from the perspective of BPG based on the public dataset [MTS15, HM16]³. First, co-purchase records between two products may

³Dataset is available at <https://jmcauley.ucsd.edu/data/amazon/>

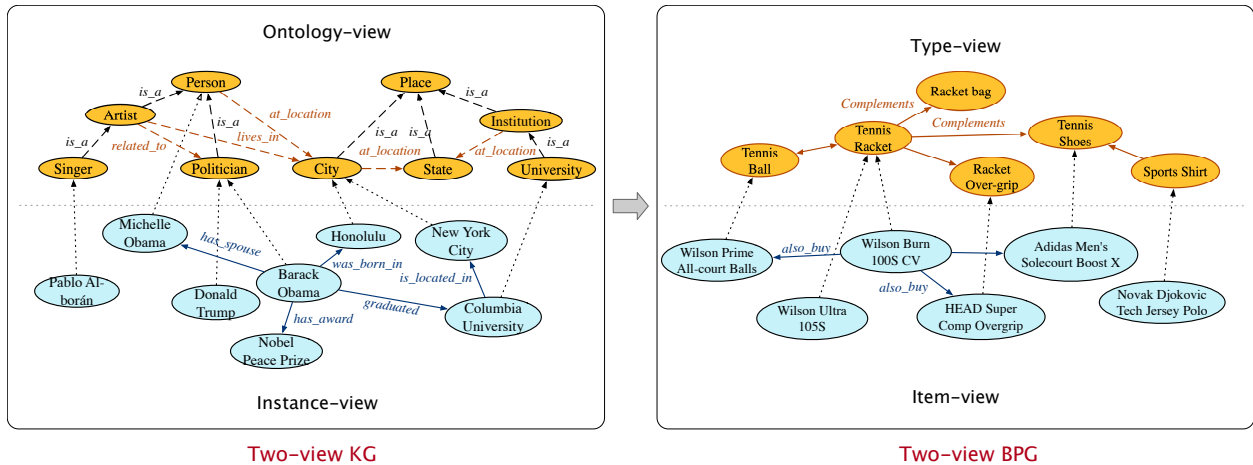


Figure 4.4: Connections between the two-view KG and the two-view BPG.

Table 4.2: Comparison between KGs and BPGs.

Property	KGs	BPGs
Source	Established facts	Product catalog and use-product interaction
Quality	Observed facts are mostly well-established and plausible.	Much noisier, especially for user behavior data
Relations	Typically hundreds of relations in a general KG, such as <code>born_in</code> , <code>director_of</code> , etc	A few relations (< 10) defined from user behavior, such as <code>also_view</code> and <code>also_bought</code> .
Attributes (common)	Entity types, numerical features, descriptions, and many other additional features	
Attributes (difference)	Different attributes apply due to the heterogeneity of entities. For example, <code>birth_dates</code> for person entities while <code>has_altitude</code> for location entities.	Each product typically has same attributes, such as images, description, reviews rate scores and price, etc.
Downstream tasks	Knowledge completion, relation extraction, question answering, etc.	Recommendation, searching, etc.

not imply a complementary relationship and it is highly possible that two products can be observed in *co-view* and *co-purchase* (overlap); that is, pairs of products can be connected with multiple relations. Therefore, it is not accurate to directly consider *co-purchase* pairs as “complements”. Around 1/5 of *co-purchase* records in \mathcal{B}_{cp} are also observed in *co-view* \mathcal{B}_{cv} records, as shown in Figure 4.5, which make it difficult to classify the complementary relationship between items by simply referring to co-purchase records.

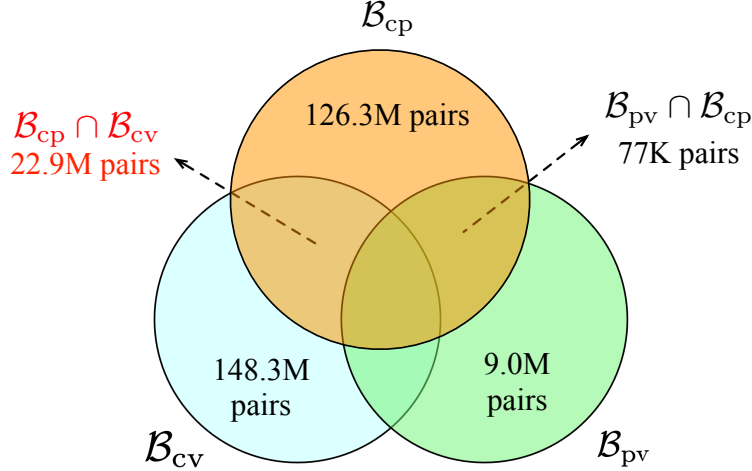


Figure 4.5: Behavior based item relations have overlaps with each other. The overlaps of $\mathcal{B}_{cp} \cap \mathcal{B}_{cv}$ no doubt cast a challenge of complementary label correctness.

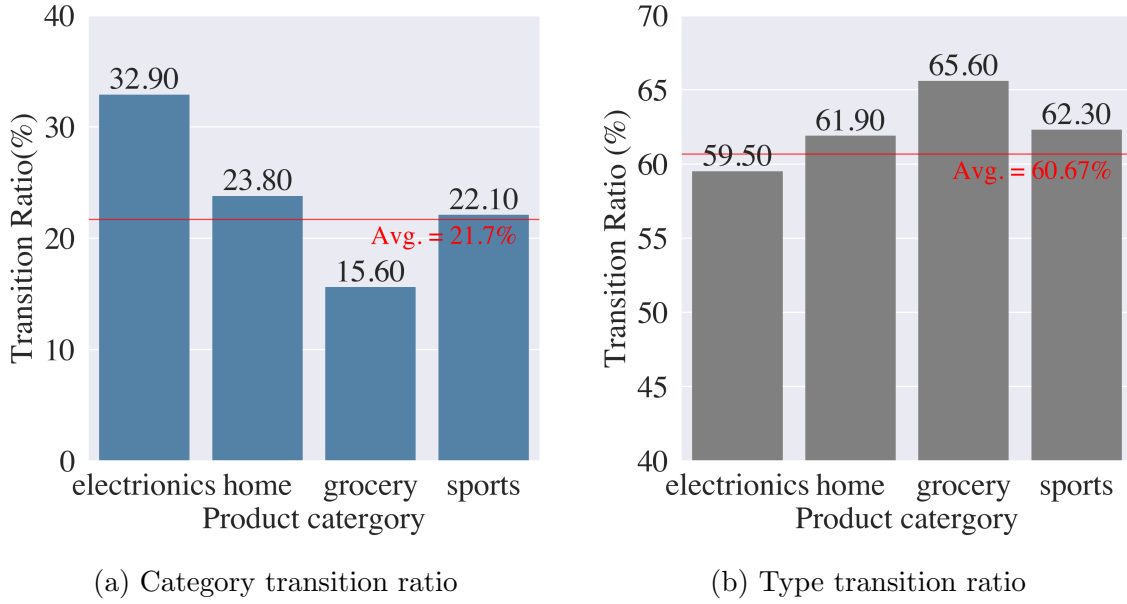


Figure 4.6: High complement transition ratio among co-purchase pairs in \mathcal{B}_{cp} in terms of categories and types.

Though it is difficult to identify item relation based on raw \mathcal{B}_{cp} , \mathcal{B}_{cv} and \mathcal{B}_{pv} data, we observe from human labeled data (shown in Table 4.3) that $(\mathcal{B}_{cv} \cap \mathcal{B}_{pv}) - \mathcal{B}_{cp}$ is a strong indicator of substitute relation with average accuracy of 90.8%. Also, $\mathcal{B}_{cp} - (\mathcal{B}_{pv} \cup \mathcal{B}_{cv})$ can be viewed as complements with reasonable confidence. Therefore, we later use the latter as purified co-purchase data in experiments.

As for the second observation, the complement relation among products is often observed in multiple categories or types. Previous research focuses on recommendation within one specific category. However, we observe a significant percentage of “complements” from historical co-purchase data, where categories or types of these product pairs differ. Figure 4.6a shows that *co-purchase* pairs often come from different categories. For example, 32.93% of “electronics” products are purchased together with “home improvement”, “office product”, etc. Even within one specific product category, it is common for two items with different functional types are observed in co-purchase record and considered as complements, which accounts for approximately 60% in the record, as shown in Figure 4.6b. This suggests multi-category dataset should be better used in complementary evaluation.

Table 4.3: Product item relationship analysis on combinations of \mathcal{B}_{cp} , \mathcal{B}_{cv} and \mathcal{B}_{pv} in terms of classification accuracy by human evaluation. on Electronics (Elec.), Grocery (Gro.) and all categories (All).

Relationship	$(\mathcal{B}_{cv} \cap \mathcal{B}_{pv}) - \mathcal{B}_{cp}$			$\mathcal{B}_{cp} - (\mathcal{B}_{pv} \cup \mathcal{B}_{cv})$		
	Electronics	Grocery	All	Electronics	Grocery	All
Substitutable	87.36	89.59	90.80	29.26	29.76	35.08
Complementary	11.31	10.01	8.45	46.33	48.47	43.17
Irrelevant	1.50	0.40	1.12	24.40	21.76	21.72

4.2.4 Problem Definition: Complementary Recommendation

Given the input as *catalog features* \mathcal{C} (including item type w) and customers behavior data \mathcal{B} , for a query item i , we recommend a set of items S_i , aiming at optimizing their co-purchase probability $\mathbb{P}_{cp}(\{i, j\}), j \in S_i$ and recommendation diversity as well, by finding the parameters Θ (item or type embeddings, transition models, etc). All the notations used in this paper are summarized in Table 4.4.

4.3 Modeling

In this section, we present the main algorithm of P-Companion for product complementary recommendation. P-Companion model is an embedding based, joint training, end-to-end

Table 4.4: P-Companion notations used in Chapter 4

Symbol	Description
$i \in \mathcal{I}$	Item i in item set \mathcal{I}
w_i	Item type of i
θ_i	Product2vec embedding vector for item i
ϕ_{w_i}	Embedding vector for query type w_i
$\phi_{w_i}^c$	Embedding vector for complementary type w_i
$\theta_i^{w_c}$	Item i 's projected embedding vector based on type w_c
γ_{w_i}	Complementary base vector predicted for type w_i
$y_{i,j}$	Binary label to indicate item i and j 's relationship
$z_{i,j}$	Attention weight given item i and its neighbor item j
$\{W^{(k)}\}, \{b^{(k)}\}$	Learnable weight matrices and bias vectors
α	Trade-off parameter to control complementary type transition and type-item prediction
\mathcal{T}	All the item pairs used for model training
$\lambda, \lambda_i, \lambda_w, \epsilon$	different margin parameters in loss functions

framework. Figure 4.7 shows the high-level model architecture of P-Companion. The model has the three major components:

- **Graph-based Product Representation Learning (named as Product2vec).** It encodes the graph-structured BPG and leverages item textual features and product similarities to learn product embeddings. It adapts the graph attention network (GAT) [VCC18] for effective training and serves as the foundation of P-Companion. (Section 4.3.1)
- **Complementary Type Transition.** It learns the complementary transition in item type subspace through a neural network. In other words, it can successfully learn the complementary types given one query item type. Also, we can control the diversity in recommendation by controlling the transition parameters. (Section 4.3.2)
- **Complementary Item Prediction.** As the last step, by learning a projection function between type and item space where types are associated with their corresponding items, we seek to find the best item match for one specific query item with complementary types. (Section 4.3.3)

These three components are logically interrelated. We first employ the Product2vec

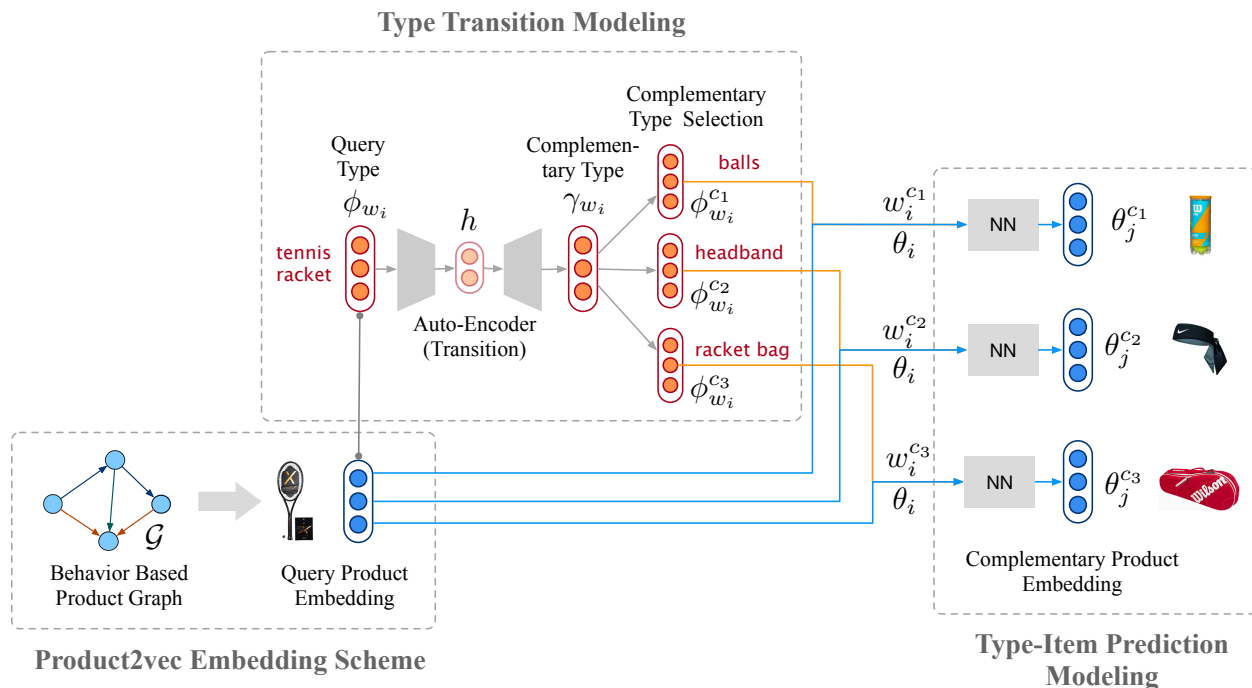


Figure 4.7: P-Companion model architecture for complementary recommendation. As an embedding based recommender, it has three major components: type transition, type-item prediction, along with product2vec for pretrained product embeddings.

to obtain pre-trained product vectors and facilitate the subsequent product complementary recommendation processes. To complete the recommendation task, P-Companion first aims at finding reasonable and diverse complementary types and then matching the product type given the type-item projection.

In addition, we also explain the joint training objective function in Section 4.3.4 and supplementary training/inference details and analysis in Section 4.3.5.

4.3.1 Product2vec: Pretrained Product Representation

Product2vec is proposed to learn embedding representations for such items that we preserve their similarities based on customer behavior data and catalog features. The learned embeddings will be used as pre-trained base representations for items, which is foundational to complementary modeling. The fundamental assumption of Product2vec is that, for pairs of items that shares similar properties and are connected in the *co-view* or *purchase-after-view*

relations, their embeddings should be close to one another in a low-dimensional latent space. Figure 4.8 shows such **Product2vec** encoder model architecture.

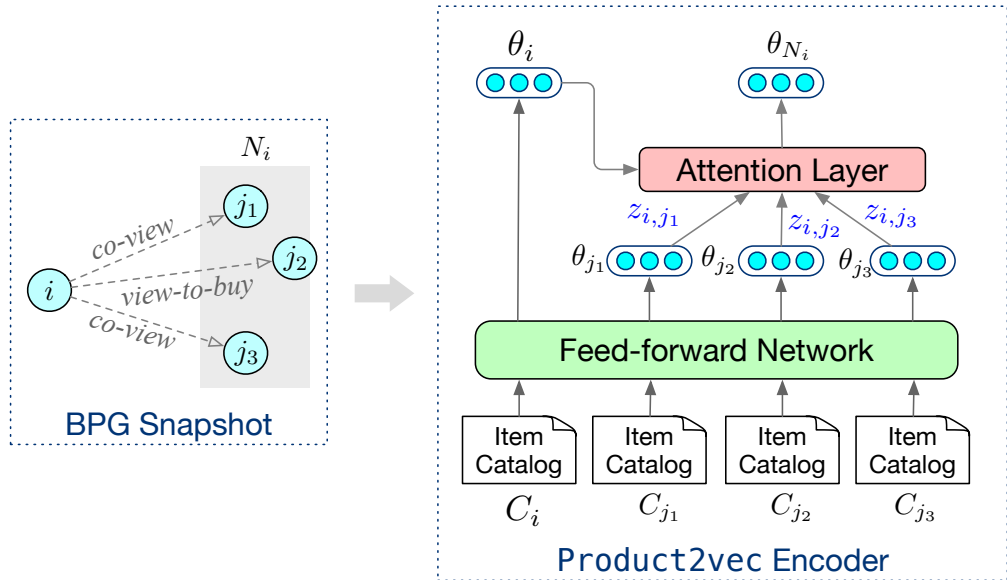


Figure 4.8: GNN-based **Product2vec** module architecture, which learns effective product embeddings given its textual features and aggregation from similar products.

Product2vec starts with taking each item’s title and category features as input and applies a feedforward network to all items i identically (sharing parameters of a three-layer $FFN(\cdot)$) and output to the k -dimensional embedding space, as shown in Equation 4.1.

$$\theta_i = FFN(C_j) = \sigma(\sigma(C_i W^{(1)} + b^{(1)}) W^{(2)} + b^{(2)}) W^{(3)} + b^{(3)} \quad (4.1)$$

where C_i is the raw feature vector for item i ’s catalog and $\sigma(\cdot) = \tanh(\cdot)$ is a non-linear activation function. $W^{(1)}, W^{(2)} \in \mathbb{R}^{d \times d}, W^{(3)} \in \mathbb{R}^{d \times k}$ are weight matrices with the corresponding biased terms $b^{(1)}, b^{(2)}, b^{(3)}$.

Through $FFN(\cdot)$, each product item i has been transformed into k -dimensional representation θ_i . The next step is GAT layers which selectively aggregates the neighbors from *co-view* relation or the *purchase-after-view* relation, which is a variant of [VCC18]. The intuition is that, since the customer behavioral data can be quite noisy in daily transactions, we expect that by the attention mechanism can help alleviate the noisy signals from these

neighbors. More specifically, given an item i and the set of neighbor items $\{j\}$ in N_i , an attention vector $z_{i,j} \in \mathbb{R}^{|N_i|}$ is calculated based on θ_i and $\{\theta_j\}$ normalized on the softmax function, which can adaptively capture the similarities when summarizing items $\{j\}$ in N_i ,

$$z_{i,j} = \text{softmax}_j (\theta_i^T \theta_j) = \frac{\exp(\theta_i^T \theta_j)}{\sum_{j' \in N_i} \exp(\theta_i^T \theta_{j'})} \quad (4.2)$$

Thus the information of item i from neighborhood aggregation N_i can be summarized as, $\theta_{N_i} = \sum_{j \in N_i} z_{i,j}$. For an item i with N_i , we mark it with a positive label $y_{i,N_i} = 1$. To conduct non-trivial model learning, for each item i , we create negative samples \hat{N}_i , as negative training instance⁴ with labels $y_{i,N_i} = -1$. Therefore, the objective function of **Product2vec** is designed to optimize a hinge loss function in Equation 4.2.

$$\begin{aligned} & \min \sum_{i \in \mathcal{I}} \left(l(y_{i,N_i}, f(i, N_i)) + l(y_{i,\hat{N}_i}, f(i, \hat{N}_i)) \right) \\ & = \min \sum_{i \in \mathcal{I}} \sum_{y \in \{\pm 1\}} \{ \max(\epsilon - y \cdot (\lambda - \|\theta_i - \theta_{N_i}\|_2^2)) \} \end{aligned} \quad (4.3)$$

where $y = \{y_{i,N_i}, y_{i,\hat{N}_i}\}$, $f(\cdot)$ is a metric function, λ is the base distance to distinguish N_i and \hat{N}_i and ϵ is the margin distance. Equation 4.3 aims at forcing the distance between θ_i and θ_{N_i} less than $\lambda - \epsilon$ while pushing θ_i far away from θ_{N_i} with at least $\lambda + \epsilon$ distance.

It is noteworthy that $FFN(\cdot)$ is optimized with the learning of Equation 4.3 and once model training is done, $FFN(\cdot)$ is extracted independently to generate embeddings for a large number of web-scale product items and such embedding repository acts as important input for subsequent complementary modeling, which is described in Section 4.3.3.

4.3.2 Complementary Type Transition

P-Companion takes pairs of query product items $\{i\}$ and candidate complement items $\{j\}$ along with their types $\{(w_i, w_j)\}$ and co-purchase label $\{y_{i,j}\}$ as input. Complementary type transition only take such type transition pairs $\{(w_i, w_j)\}$ into consideration, seeking to build

⁴Details are explained in Section 4.3.5

up a model complementary type transition $w_i \Rightarrow w_j$, which benefits the recommendation quality by focusing on predicting accurate complementary types before item-level prediction.

For each type w , we assign two learnable embedding vectors $\phi_w, \phi_w^c \in \mathbb{R}^L$ to it, indicating its context position as query type or complementary type. Given a pair of items (i, j) with types w_i and w_j respectively, we use an encoder-decoder module to transform ϕ_{w_i} , the query embedding vector of w_i , to its complementary base vector γ_{w_i} , which will be used to predict complementary types for w_i . An encoder-decoder architecture is shown in Equation 4.4 and 4.5:

$$h = \text{Dropout} \left(\text{ReLU} \left(\phi_{w_i} W^{(4)} + b^{(4)} \right) \right), \quad (4.4)$$

$$\gamma_{w_i} = h W^{(5)} + b^{(5)}, \quad (4.5)$$

where $W^{(4)} \in \mathbb{R}^{L \times \frac{L}{2}}$ and $W^{(5)} \in \mathbb{R}^{\frac{L}{2} \times L}$ are weight matrices for encoding and decoding types. Then, we optimize the relationship between the predicted type γ_{w_i} and ground-truth type $\phi_{w_j}^c$ with the label $y_{i,j}$ by using the hinge loss function in Equation 4.6.

$$\min \sum_{i,j \in \mathcal{T}} \left(\max \left\{ 0, \epsilon_w - y_{i,j} \left(\lambda_w - \|\gamma_{w_i} - \phi_{w_j}^c\|_2^2 \right) \right\} \right), \quad (4.6)$$

where λ_w is the base distance to distinguish γ_{w_i} and $\phi_{w_j}^c$, ϵ_w is the margin distance. Equation 4.6 aims at forcing the distance between γ_{w_i} and $\phi_{w_j}^c$ to lower than $\lambda_w - \epsilon_w$ when $y_{i,j} = 1$ while pushing $\phi_{w_j}^c$ far away from γ_{w_i} with at least $\lambda_w + \epsilon_w$ distance when $y_{i,j} = -1$.

As a summary, P-Companion uses the neural network based type transition and satisfies the requirement of asymmetric modeling (C2) for complementary recommendation.

4.3.3 Complementary Item Prediction

In this section, we propose a learning approach to transfer the different complementary type embedding to corresponding items, as a type-item projection. As Figure 4.7 describes, by using *Complementary Type Transition* module to transfer query type embedding ϕ_{w_i} to its complementary base embedding γ_{w_i} , we design an item-embedding transition module that

takes advantage of the predicted type embedding vector γ_{w_i} to project the original item embedding θ_i to different complementary subspaces via Equation 4.7.

$$\begin{aligned}\theta_i^{w_c} &= \theta_i \odot (\phi_{w_c}^c W^{(6)} + b^{(6)}), \\ s.t., \quad &\|\phi_{w_c}^c - \gamma_{w_i}\|_2^2 \leq \beta,\end{aligned}\tag{4.7}$$

where $W^{(6)} \in \mathbb{R}^{L \times d}$, \odot represents element-wise product and β is the similarity threshold to determine which complementary types will be used to recommend complementary items. We can also explicitly set how many complementary types for each query type, as the implementation in our experiment. Based on different complementary type embeddings $\{\phi_{w_c}^c\}$ that are close enough to γ_{w_i} , we can transfer item i 's embedding θ_i to multiple complementary targets $\{\theta_i^{w_c}\}$. For each of complementary candidate j with its type w_j , we still use a hinge loss to optimize the objective function based on $\theta_i^{w_c}$, θ_j and label $y_{i,j}$ according to Eq. (4.8). Here w_c is selected based on β and γ_{w_i} .

$$\min \sum_{i,j \in \mathcal{T}} \max \{0, \epsilon_i - y_{i,j} (\lambda_i - \|\theta_i^{w_c} - \theta_j\|_2^2)\}.\tag{4.8}$$

With type-item modeling as projection, combined with type transition in Section 4.3.2, **P-Companion** is not simply a similarity-based recommender (**C2**) but can also provide reasonable recommendation on the product level (**C3**). More importantly, by controlling the number of different complementary types (or selection threshold), we can successfully achieve the diversified recommendation (**C5**), comparing to existing methods.

4.3.4 Joint Training

In model training phrase, **P-Companion** jointly optimizes complementary type and item objective functions based on Equation 4.6 and Equation 4.8. To strengthen the connection between two objective functions, for each training instance, we force ϕ_{w_c} to be the same as γ_{w_i} in Equation 4.7. Once the model is well-trained, **P-Companion** sticks with Equation 4.7 to predict complementary items based on different complementary types. Therefore, the

overall objective function can be written as Equation 4.9.

$$\begin{aligned} \min \sum_{i,j \in \mathcal{T}} \alpha & \left(\max \left\{ 0, \epsilon_i - y_{i,j} \left(\lambda_i - \|\theta_i^{w_j} - \theta_j\|_2^2 \right) \right\} \right) \\ & + (1 - \alpha) \left(\max \left\{ 0, \epsilon_w - y_{i,j} \left(\lambda_w - \|\gamma_{w_i} - \phi_{w_j}^c\|_2^2 \right) \right\} \right), \end{aligned} \quad (4.9)$$

where α is a hyper-parameter to control the tradeoff between complementary type modeling and complementary item modeling.

4.3.5 Training and Inference

In this section, we provide more details on P-Companion training as well as model analysis.

Pretrained Type Representations from Transition Matrix Although P-Companion is an end-to-end solution for diversified complementary item recommendation, rather than randomized initialization, We extract a type transition matrix \mathcal{M} from co-purchase data where each entry $\mathcal{M}[w_i, w_j]$ in \mathcal{M} records the number of co-purchase with w_i as query type and w_j as complementary type. Based on each entry $\mathcal{M}[w_i, w_j]$ in \mathcal{M} , we follow Equation 4.4 and Equation 4.5 to obtain two vectors, γ_{w_i} and $\phi_{w_j}^c$, then we optimize Equation 4.10 to fit $\mathcal{M}[w_i, w_j]$.

$$\sum_{w_i, w_j} \left(\sigma \left(\gamma_{w_i}^T \phi_{w_j}^c \right) - \sigma \left(\mathcal{M}[w_i, w_j] \right) \right)^2, \quad (4.10)$$

where $\sigma(\cdot)$ is the sigmoid function that is used to normalize and rescale similarities of w_i and w_j as well as $\mathcal{M}[w_i, w_j]$. Once type transition pretraining is completed, we use the parameter values to initialize corresponding parameters in Equation 4.9 and then perform joint learning on complementary type and item recommendation.

Model Inference As introduced in Figure 4.7, given one query item i (with its embeddings θ_i by Equation 4.1) and associated type w_i (with embeddings ϕ_{w_i} , we can predict its complementary types w_i^c (embedding $\phi_{w_i}^c$) through Equation 4.4 and 4.5 and thus obtain its complementary item recommendation $\theta_i^{w_i^c}$ through Equation 4.7. We can control the number of recommendations for query items by assigning the number of complementary type selection and complementary item per type candidate.

Negative samples For `Product2vec` training (in Equation 4.3), we use $(\mathcal{B}_{cv} \cap \mathcal{B}_{pv}) - \mathcal{B}_{cp}$ as positive examples and $\mathcal{B}_{cp} - (\mathcal{B}_{pv} \cup \mathcal{B}_{cv})$ as negative ones; for `P-Companion` training (in Equation 4.9), we consider frequent co-purchase data in as positive samples and $(\mathcal{B}_{cv} \cap \mathcal{B}_{pv}) - \mathcal{B}_{cp}$ as negative. This is according to our observations in Table 4.3. The negative sample ratio is fixed as 1.0 in subset datasets (i.e. electronics, grocery, and furniture datasets, see Section 4.4.1 while we use all data into training for the All-Group dataset.

Other training settings The (hyper)parameters are set as follows: product embedding dimension $d = 128$, pretrained item type embedding dimension $L = 64$, margin parameters $\lambda = 1.0$ and $\epsilon = 1.0$. Tradeoff parameter between type transition and item prediction is set as $\alpha = 0.8$ (see ablation study in Section 4.4.5).

Complexity analysis Our proposed `P-Companion` including the product training module named `Product2vec` are scalable to large-scale online e-commerce platforms. In terms of model parameter size, `Product2vec` contains embedding table for all products, which is $\mathcal{O}(N_p d)$ (N_p denotes the total number of product items) and product encoder parameters of size $\mathcal{O}(d^2 + dk)$. `P-Companion` takes the type embedding table of size $\mathcal{O}(N_t L)$ (N_t denotes the total number of types) with the type transition neural network module of the parameter size as $\mathcal{O}(L^2)$ and type-product prediction network of the parameter size as $\mathcal{O}(Ld)$. The parameter size allows scalable implementation for web-scale product recommendation. As for training time complexity, `P-Companion` and `Product2vec` is approximately linear proportional to the number of observed pairs in the BPG, that is, $\mathcal{O}(n_e(\mathcal{B}))$ where n_e denotes the number of edges in the graph \mathcal{B} (typically $n_e(\mathcal{B}) \gg d, k$). It is worth mentioning that inference can be done offline to obtain the recommendation list for each product and directly fetched from the stored output from `P-Companion`, which is even more time-efficient.

4.4 Experiments

In this section, we perform an extensive set of experiments on product data from a leading e-commerce service. We first describe our dataset and baselines in Section 4.4.1. Next,

we present quantitative results on recommendation tasks on historical frequent co-purchase data and compare P-Companion with baselines. To overcome the incompleteness of historical data, we also perform human evaluation from MTurk. Finally, we include case studies to understand how P-Companion outperforms previous approaches as well as ablation study on the trade-off hyperparameter α in Section 4.4.5.

4.4.1 Experiment Setup

Dataset We evaluate P-Companion on a real-world dataset obtained from one leading e-commerce service platform, which includes a large number of product catalog features and customer behavioral data, with the same pattern as product metadata in [MTS15, HM16, MPL15]. We select two specific product category groups as subsets: electronics, grocery, and the dataset with all categories. Statistics about all these three datasets are summarized in Table 4.5.

Table 4.5: Dataset statistics.

Dataset	Electronics	Grocery	All-Groups
# Product items	97.6K	324.2K	24.54M
# Types	5.6K	6.5K	34.8K
# \mathcal{B}_{cp} pairs	130.6K	804.1K	62.16M
# \mathcal{B}_{cv} pairs	3.15M	8.96M	1,154M
# \mathcal{B}_{pv} pairs	325.1K	1.105M	83.75M

Note that the subsets are selected by the query product category; however, the candidate products in the complement pairs do not necessarily belong to the same category. That is, we allow products in different categories as long as it is related by any query product, which is consistent with our claim in Section 4.2.

Baselines We compare P-Companion with the following state-of-the-art baseline approaches. We use the default setting on hyperparameters for these baselines. Note that we adopt the default setting on these baseline methods.

- **Co-purchase (CP)** As the most straightforward way, we can directly output the items in the co-purchase records for complementary recommendation. Co-purchase pairs (query

item i , co-purchase item j) can be extracted from massive purchase customer data and naturally such pairs form an asymmetric relation between i and j .

- **Sceptre** [MPL15] This approach utilizes topic modeling on item textual features (review text) and logistic regression for substitute/complement classification. Category information is also applied with a sparse encoding technique.
- **PMSC** [WJR18] Each product item has its source embedding and target embeddings for query and candidate contexts. It adopts additional relation-aware parameters to model multiple item relations and later feed in a neural network for classification.
- **JOIE** [HCY19] Originally designed for two-view knowledge graph embeddings and not targeting at solving the complementary product recommendation, we can adapt JOIE to product-type views in BPG instead of entity-concept views in KG and consider complementary as a triple completion task, that is, to infer the triplets (*Entity: Query product, Relation: co-purchase, Entity: ?*) from the learned embedding and regularization from product-type cross-view association.

It is noteworthy to mention that, in Section 4.4.2, co-purchase history data are intentionally used for evaluation, which is widely adopted as other previous work.

4.4.2 Historical Co-Purchase Evaluation

In real-world applications, only a limited number of complements can be recommended because of the space limitation. Thus rather than distinguishing complements or non-complements between a given pair of products [MPL15, WJR18], we directly evaluate by ranking basis, where we aim that good complements are scored higher than irrelevant ones. In other words, given a query product i from the frequent co-purchase pairs (i, j) , we provide a list of top- K recommendations S_k and evaluate the **P-Companion** as well as all baselines by checking whether the model can successfully predict the corresponding target

Evaluation metrics A standard measurement for ranking tasks is the Hit@ K score. Given a pair of items (query item i , co-purchased item j) in co-purchase test data, the Hit@ K score

is defined as,

$$\text{Hit}@k = \begin{cases} 1, & \text{if } j \in S_k \\ 0, & \text{else} \end{cases}, \quad k = 1, 3, 10, \dots$$

where S_k is the k -element list of recommendations from the model. We report both Hit@ K scores on both item level and type level (if applicable). That is, for JOIE and P-Companion, we first predict the top- K complementary types and see whether the model can successfully predict the correct item type of j . As for the item level, we target at the ability to predict the exact Since co-purchase data are used as ground truth for evaluation, we compare P-Companion with Sceptre, PMSC, and JOIE, which are introduced in Section 4.4.1.

To validate the effect of diversity in recommendation, we also experiment on different settings of P-Companion. More specifically, we recommend a total number of 60 products, however, the number of recommended types and number of items for each type differs. We test P-Companion in the following four recommendation settings separately during the inference stage: to recommend only 1 type and 60 items for the type (denoted as “1 type \times 60 items”) together with “3 types \times 20 items”, “5 types \times 12 items” and “6 types \times 10 items”. At the same time, all baselines output 60 items as recommendations, which is the same as P-Companion.

Table 4.6: Results of complementary recommendation based on historical FCP records on *Electronics* dataset.

Datasets	Electronics					
Level	Item Hit score			Type Hit score		
Metrics	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
Sceptre	0.069	0.079	0.101	n/a	n/a	n/a
PMSC	0.112	0.135	0.169	n/a	n/a	n/a
JOIE	0.141	0.164	0.181	0.095	0.190	0.304
P-Companion	0.145	0.170	0.187	0.113	0.206	0.348

Results We report the results shown in Table 4.6 for the Electronics and Grocery subgroups. P-Companion outperforms all baselines in the item level prediction with an average relative gain of 4.2% compared to the strongest baseline. In terms of type level, Comparing to JOIE with the item-type view, P-Companion improves by 9.9% on the Electronics dataset and by

Table 4.7: Results of complementary recommendation based on historical FCP records on *Grocery* datasets.

Datasets	Grocery					
Level	Item Hit score			Type Hit score		
Metrics	Hit@1	Hit@3	Hit@10	Hit@1	Hit@3	Hit@10
Sceptre	0.018	0.032	0.040	n/a	n/a	n/a
PMSC	0.024	0.053	0.087	n/a	n/a	n/a
JOIE	0.026	0.058	0.099	0.079	0.170	0.281
P-Companion	0.030	0.063	0.104	0.083	0.177	0.293

Table 4.8: Results of complementary recommendation based on historical FCP records on large-scale All-Group dataset. (H@k denotes Hit@k score.)

Datasets	All Category Groups					
Level	Item Hit score			Type Hit score		
Metrics	H@1	H@3	H@10	H@1	H@3	H@10
Sceptre	0.019	0.041	0.059	n/a	n/a	n/a
JOIE	0.037	0.062	0.104	0.054	0.153	0.204
P-Companion	0.037	0.068	0.108	0.064	0.161	0.212

Table 4.9: Performance of P-Companion with different number of predicted item types on Electronics and Grocery dataset.

Dataset		Electronics	Grocery
Model & Setting		Hit@60	Hit@60
Sceptre		0.124	0.085
PMSC		0.179	0.139
JOIE		0.200	0.155
P-Companion	1 type \times 60 items	0.138	0.088
	3 types \times 20 items	0.198	0.153
	5 types \times 12 items	0.222	0.189
	6 types \times 10 items	0.227	0.187

4.5% on the Grocery dataset. In Table 4.8, we observe similar phenomena on larger “All-Group” dataset with a relative 3.8% increase on item-level Hit score on average and increase on type-level against JOIE.⁵ We believe this is due to the following reasons, (i) **P-Companion** infers complementary products by targeting the complementary type first rather than only modeling product relationships in Sceptre and PMSC. Item types can be considered as functionality abstraction and enable more accurate recommendation; (ii) Comparing with the similar item-type view model in JOIE, **P-Companion** type transition module can better capture the complementary relations between types. Also, **P-Companion** explicitly involves both query item and predicted complementary types for the item-level recommendation, but JOIE uses query items only during inference.

As for the effect in diversified complementary modeling, we show the results in Table 4.9 to validate the benefit of diversified recommendations. The best **P-Companion** variant outperforms the best baseline model by a hit-score increase of 0.027 on Electronics (6 types) and 0.034 on Grocery (5 types). However, if the number of recommendation types is restricted, the performances of **P-Companion** drop significantly. Moreover, **P-Companion** variant with only 1 type in recommendation can only slightly outperform with the Sceptre, but not PMSC or JOIE. In summary, though the same number of items are recommended, **P-Companion** manages to provide a diverse recommendation explicitly and results in better item-level hit score. Also, **P-Companion** is capable of control the diversity in recommendation regarding different categories of products. Inspired by the observation, we may reasonably recommendation with more types in complement recommendation list for Electronics than Grocery, based on the different natures of products.

4.4.3 MTurk Evaluation

Though co-purchase data can provide complement products with reasonable confidence and serve as ground truths in Section 4.4.2, we also observe that such co-purchase dataset is relatively sparse considering the scale of all products and far from being complete. In

⁵Due to the scalability issue, the results of PMSC is not available.

other words, it fails to include all possible truths on complements and lots of cases that qualifies complements cannot be successfully identified due to the absence in co-purchase history data. To avoid incompleteness in co-purchase data and provide a more thorough understanding of the results, we involve human evaluation on such complementary recommendation tasks by launching MTurk⁶ surveys to collect feedback on the recommendation results. Rather than using incomplete co-purchase data as ground truths, we compare the quality P-Companion recommendation with co-purchase data, in terms of recommendation relatedness and coverage.

MTurk settings & evaluation metrics We design surveys based on our model recommendations by the following steps. We first sample top-1000 glance-viewed items as queries and select the top-15 recommendations from P-Companion for these items (3 types \times 5 items per type). When preparing item pairs for MTurk, we interleave recommendations from top 3 complementary types following the predicted complementary type order as well as complementary item prediction order within the scope of each type ⁷.

As shown in Figure 4.9 as questionnaire snapshot is shown in, for each recommended pairs of products, we ask 5 different MTurk labelers about the question that *“Given you decide to purchase the base product, would you be interested in purchasing the recommended product together with the base product?”* and prepare the following different answers,



⁶MTurk is a crowdsourcing website for businesses to hire remotely located “crowd workers” to perform discrete on-demand tasks that computers are currently unable to do. Website: <https://www.mturk.com>.

⁷More specifically, given a cell phone as a query item, our model first predicts top 3 complementary types as phone case, screen protector and charger. Within each complementary type, we collect the top 5 ASIN recommendations. Then based on the ranked complementary types and ASINs, our final recommendations will be arranged by such multiple groups.

Instructions

- Given you decide to purchase the base product, would you be interested in purchasing the recommended product together with the base product?
- You should focus on the functionality and please consider whether the recommended product inspires you potential needs as complementaries/supplementaries.
- Some HITs contain sanity-check questions which have obvious answers. If your answers are wrong on these questions, your work may be rejected.
- You can click on the product image or title to navigate to the detail page at Amazon.

Question 1: Given you decide to purchase the base product, would you be interested in purchasing the recommended product together with the base product?

Base Product	Recommended Product	Answer
 <p>Polk Audio Atrium 4 Outdoor Speakers with Powerful Bass (Pair, White) All-Weather Durability Broad Sound Coverage Speed-Lock Mounting System</p>	 <p>Polk Audio Atrium 4 Outdoor Speakers with Powerful Bass (Pair, Black) All-Weather Durability Broad Sound Coverage Speed-Lock Mounting System</p>	<input type="radio"/> Yes, I am very likely to buy them together. <input type="radio"/> The recommendation inspires me the potential needs to purchase, but not this right one. <input type="radio"/> No, the recommendation is relevant but I am less likely to buy them together. <input type="radio"/> I do not think they are relevant.

Question 2: Given you decide to purchase the base product, would you be interested in purchasing the recommended product together with the base product?



Base Product	Recommended Product	Answer
 <p>Apple 13 Inch MacBook Pro / MD101LL/A / 2.5GHz Intel Core i5, 4GB RAM, 500GB HDD, Intel HD 4000 Graphics, DVDRW, WIFI Wireless, iSight Webcam</p>	 <p>MOSISO Laptop Sleeve Compatible 2018 MacBook Air 13 A1932 Retina Display/MacBook Pro 13 A1989 A1706 A1708 USB-C 2018 2017 2016/Surface Pro 6/5/4/3, Polyester Bag with Vertical Pocket, Wine Red</p>	<input type="radio"/> Yes, I am very likely to buy them together. <input type="radio"/> The recommendation inspires me the potential needs to purchase, but not this right one. <input type="radio"/> No, the recommendation is relevant but I am less likely to buy them together. <input type="radio"/> I do not think they are relevant.

Figure 4.9: One MTurk survey snapshot for complements recommendation evaluation. MTurk workers are asked to tell their purchase willingness in a range of 0-3.

1. Yes, I am very likely to buy them together. (**Score 3, perfect**)
2. The recommendation inspires me the potential needs to purchase, however just not the right one. (**Score 2, inspiring**)
3. No, the recommendation is relevant but I am less likely to buy them together. (**Score 1, relevant**)
4. I do not think they are relevant. (**Score 0, failed**)

We believe the Score-3 answer will indicate the co-purchase probability for our recommendations. As mentioned before, the product-level complementary recommendation is very challenging, so we design the second answer to collect compromising feedback about purchase probability by using the Score-1 and Score-2 answers. As for the evaluation metrics we expect to measure the recommendation relevance, or say *relevance rate* as one evaluation metric, defined as the percentage of all Score 1-3 answers among all the collected feedback data. We also calculate the *average score* of all collected data points as an alternative measurement of recommendation relevance. Finally, from another perspective of the recommendation task, we also compare the *item coverage*, defined as *the total number of product items that show up in all recommendations* given all the query items in the dataset.

Table 4.10: MTurk comparison between P-Companion’s Top-5 recommendations and co-purchase record. Percentage of Score X represents the proportion of pairs labeled with Score-X.

Model	Co-Purchase	P-Companion				
		Pos-1	Pos-2	Pos-3	Pos-4	Pos-5
% of Score 3	0.46	0.43	0.43	0.42	0.45	0.42
% of Score 2	0.25	0.27	0.27	0.27	0.26	0.27
% of Score 1	0.27	0.27	0.26	0.26	0.27	0.26
% of Score 0	0.02	0.02	0.04	0.04	0.03	0.04
Rel. Rate	0.98	0.97	0.96	0.95	0.97	0.96
Avg. Score	2.15	2.12	2.09	2.07	2.13	2.08

Table 4.11: Query item coverage comparison on all three datasets. P-Companion shows better product item coverage over co-purchase record.

Dataset	Model	#Queries	#Recom. Items	#Average
Electronics	Co-purchase	64.1K	130.6K	2.04
	P-Companion	97.6K	1.464M	15.0
Grocery	Co-purchase	278.9K	804.1K	2.88
	P-Companion	324.2K	4,863.7K	15.0
All-Group	Co-purchase	23.90M	62.16M	2.60
	P-Companion	24.54M	368.1M	15.0

Results Product recommendation relatedness and coverage evaluation are shown in Table 4.10 and Table 4.11 respectively. From the results, P-Companion achieves over 95% relevance rate and 40% co-purchase rate. More examples are introduced in Section 4.4.5 as case study.

We can observe that although CP has a slightly better average MTurk scores (2.15 v.s. 2.12), P-Companion significantly improves item coverage in recommendation to around 6.4 times larger than CP. This results from the fact that P-Companion infer diverse complements by complementary type transition and type-item project, while co-purchase history entirely relies on observed purchase patterns without any generalization ability. The increase in item coverage can potentially bring important business values in real-world applications since high-quality complementary recommendation does not require history customer browsing data as requisite, which is generally difficult for cold-start items.

4.4.4 Production Deployment

We also launch our P-Companion on the online e-commerce platform and compare with the previous recommendation setting which mostly based on history CP record via A/B Testing.

Deployment Setting We deploy a stable pipeline to generate complementary recommendation datasets for online services. Hence, we create and run the following two A/B testing experiment to compare the performance of CP record (“control group”) and P-Companion (“experimental group”) recommendation candidates:

- For CP’s current covered key items, the control group will be the recommendations from CPrecord. As for the experimental group, we have the following three treatments,
 - Recommend items with the most likely predicted complementary type. Items are generated based on both model score and business rules, e.g., highest review scores, lowest price, best sales, etc.
 - Recommend items from multiple (most likely 3) predicted complementary types. From each predicted complementary type, we select ASINs based on model score and business rules.
 - Recommend items from CP-generated complementary types and will append by model predicted complementary types if there are not enough complements from CPtypes.
- For those items that CP currently cannot cover, the control group will be no-show of the

Co-purchase Suggestions widget, and the experimental group will still be the above three.

For evaluation metrics, compared to the control group, we report the percentage of sales revenue increase and product recommendation coverage (i.e. the total percentage of product that has been enabled with complements for recommendation) for different categories, which represents the business contribution of P-Companion (experimental group) on the whole. We can see from the setting that the first test reflects the P-Companion influence on sales revenue while the second mostly validates the effect of recommendation coverage.

4.4.5 Case Study

We provide two case studies for complementary recommendation on cold-start items, type transition examples and effects of hyperparameter α as ablation study.

Recommendation on cold-start/low-resource items We compare the performance of complementary recommendations on low-resource or even cold-start items from the output of P-Companion and CP data in this case study. Cold-start items are defined as these products that have general catalog features (such as title, descriptions, types) but have limited observed relations⁸ or even no relations with other products. Because P-Companion utilizes product embeddings generated from Product2vec which takes both item features and relation with other items for embedding generation, we can still make reasonable complement recommendation from the features of the product itself and/or related “neighbors” products in BPGs. In contrast, Sceptre is mostly based on review text and PMSC purely relies on the item category and logic path constraints in BPGs and therefore without a comprehensive model capability on BPGs, their recommendation abilities for many low-resource items are hindered.











We compare the product and type hits scores as quantitative measurements with baselines on these selected cold-start items under Electronics category in Table 4.12, with the same evaluation method stated in 4.4.2. Still, P-Companion outperforms all baselines on

⁸In this case study, we select the items with less than 2 occurrences in the dataset for cold-start test purpose.

Table 4.12: Results on complementary recommendation on cold-start product items under electronics categories. (H@k denotes Hit@k score.)

Datasets	Electronics (<i>only cold-start items in testing</i>)					
Level	Item Hit score			Type Hit score		
Metrics	H@1	H@3	H@10	H@1	H@3	H@10
Sceptre	0.049	0.065	0.081	n/a	n/a	n/a
PMSC	0.073	0.093	0.111	n/a	n/a	n/a
JOIE	0.107	0.136	0.157	0.061	0.138	0.220
P-Companion	0.115	0.147	0.168	0.073	0.158	0.244

Table 4.13: Examples of complementary recommendation on cold-start items with P-Companion output. Recommendations are highly related and diverse even though the resource of co-purchase history is limited or unavailable.

Category	Query Item	Co-Purchase	Top-5 Recommendations from P-Companion
Electronics			
Grocery			
All-Group (Pet home)		None	
All-Group (Fishing tools)		None	

low-resource items on Electronics dataset with an average relative increase of 7.0% on the item level and 10.9% on the type level. More specifically, Table 4.13 shows the example recommendation results on some cold-start items, which are generally with reasonably quality. For one query item on a pet house, animal bowls and animal toys can be seen in the recommended item list.

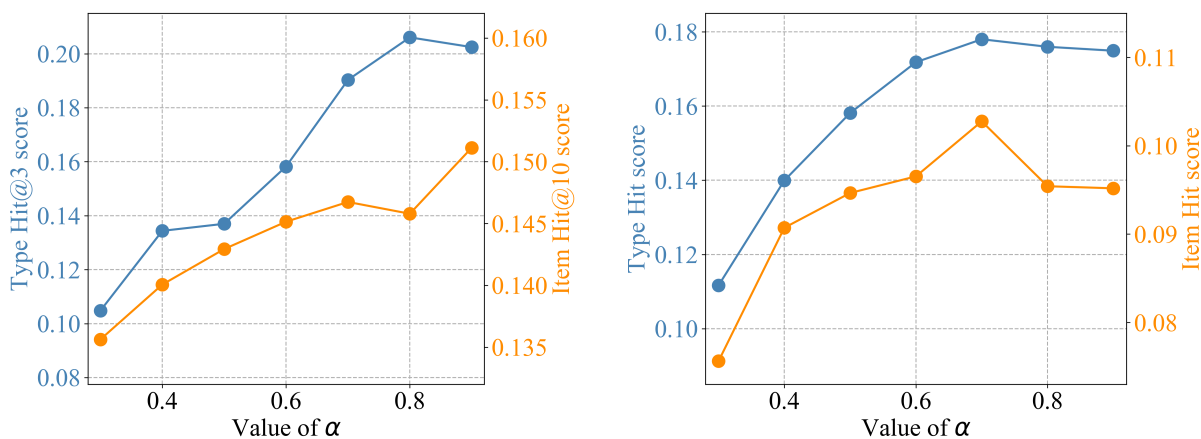
Examples of type transitions As one highlight of P-Companion, transitions between types are explicitly learned and indicate complementary relation between item types. Besides the enhanced performance of product-level recommendations, we can also extract the useful item type complementary pattern, which would contribute to better explainability of derived complementary recommendation. Table 4.14 provides examples that show reasonable transition

patterns learned from P-Companion.

Table 4.14: Type transition examples. (Only top-3 transitions are listed for each type query.)

Query Type	Top-3 Type Recommendations
cam-poweradapt	sec-digit-card, micro-sd-card, hdmi-cable
roast-coffee-bean	fridge-coffee-cream, whole-bean, white-tea
fly-fish-line	fluorocarbon-fish-line, surf-fish-rod, fly-fish-reel

Effect of joint training hyperparameter α The parameter α balances the trade-off in training between item type transition loss and type-item prediction loss. As item-level Hit scores and type-level Hit scores plotted in Figure 4.10a and 4.10b, we observe that, as α increases from 0.3 to 0.7, the performance on type and item prediction increases dramatically. When α increases from 0.7 to 0.9, the performance does not significantly improve after reaching the peak or even slightly drop on “item level”. We hypothesize that type-item prediction loss is a relatively more important factor while item type transition loss can still contribute to better performance especially on the type-level transition prediction.



(a) Effects of α on Electronics.

(b) Effects of α on Grocery.

Figure 4.10: Comparison on Type Hit@3 and Item Hit@10 performance under different hyperparameter settings of α .

4.5 Related Work

We discuss the following lines of research work that are relevant to this chapter.

Complementary Recommendation The basic task of a recommender system is to suggest relevant items given item features and user-item behaviors. Most of them are based on collaborative filtering [SK09, KB15b, SKK01, LSY03], matrix factorization [KBV09] and neural recommendation model [ZYS19, BK16]. The goal is related to our problem but instead of rating estimation, we aim at complementary relationship discovery and motivate scalable recommendation, which existing methods have sparsely explored. The most straightforward way for complementary is based on frequent pattern mining and association rules [HCX07], however, such purely data-driven methods lack modeling learning ability for complex applications though simplicity and efficiency. Some recent works [KWM18, ZLN18, ZWN09] in this direction seek to classify whether two products are complementary (or substitute), such that we can recommend complementary products based on the user’s previous purchasing or browsing patterns. Two representative examples is Sceptre [MPL15] and PMSC [WJR18]. Sceptre uses topic modeling to extract features in product review text and employs logistic regression for item relation classification while PMSC adopts source and target embeddings and improve the performance by enforcing the logic rule constraints. They mainly operates on product level and lack diversity consideration in modeling and has limited capability on cold-start items. A comparison between P-Companion and existing representative models are listed in Table 4.1 at Section 4.1.

In terms of others applications, complementary recommendation has already been applied in apparel selection and style matching [HCT18, KKL19] with a special focus in visual modeling. It is worth noted that there is a thread of research on bundle list recommendation [ZHL14, BZS19], which aims at personalized recommendation based on user’s purchase history and can be considered as a combinatorial problem. Since we currently take user-specific data into general modeling, this line of research is out of the scope of our problem in this work.

Network Embedding and Graph Neural Networks Learning on graph-structured data has been a spotlight in the past decade [AK14]. Starting from random walk based method (DeepWalk [PAS14] and nodevec [GL16]), network embedding aims at representing nodes

as low-dimensional vector representations, preserving both network topology structure and node content information and easily perform subsequent graph analytic tasks (classification, clustering). One of the state-of-the-art approaches is to use graph neural networks [HYL17b] such as GCN [KW17], GraphSAGE [HYL17a] and GAT [VCC18]. In this work, we adopt a GAT-based model to learn product embeddings, where the intuition is to selectively aggregate the information from similar products (“neighbors”) with its attributes.

Knowledge Graphs and Product Graphs Knowledge graphs (KGs) are essentially multi-relational graphs such as Freebase, DBpedia, YAGO and many domain-specific KGs. Similar to network embedding, recent work has put extensive efforts in learning distributed representation on KG entities and relations [WMW17], which are vital to capturing the latent semantic features and support relational inferences. Representative models include TransE [BUG13], DistMult [YYH15], ComplEx [TWR16], and RotatE [SDN19]. Researchers also apply similar techniques [SKB18] in product graphs (PGs), one example in e-commerce, which models the item features as well as pairwise relations between and enable downstream applications such as similarity product searching. Also, GNNs also provide a novel option to obtain KGs [SKB18] and PGs [YHC18]. In this work, we adapt one multi-view KG embedding method [HCY19] from an entity-concept view into a product-type view and serve as one strong baseline.

Diversity Modeling As diversity is a newly-introduced but critical requirement for our complementary recommendation problem, we also list some work related to diversity modeling. For example, [AK14] employs a clustering post-process to enrich diversity for recommendation and [QCZ14] proposes a contextual bandit solution. Researchers also use Determinantal Point Processes (DPP) [KT12, WMG19] for a balance of quality and diversity rooted in the mathematical formulation. Besides recommendation, diversity has also been explored in many other areas such as searching [CZL17, WLZ16]. However, to the best of our knowledge, there is no related work that concentrates the focus on diversity in complementary recommendation.

4.6 Conclusion

In this chapter, we present **P-Companion**, a novel model for large-scale diversified principled product complementary recommendation to improve the quality of e-commerce service. From customer behavioral data and product catalog features (types, etc), **P-Companion** (Figure 4.7) first employs a GNN based **Product2vec** module to learn the universal representations of products and later design an effective and efficient transition model for the asymmetric and diversified complementary recommendation. Such a model can keep product similarity, complementary relevance as well as recommendation diversity into consideration in general. Intensive experimental evaluation from historical co-purchase data and MTurk evaluation has demonstrated the effectiveness of **P-Companion** in recommending diversified and relevant complementary items over baselines methods, as well as the ability to improve recommendation coverage significantly.

The insights can be concluded as follows: (1) Relatedness and diversity are both important to product complementary recommendation; (2) The effectiveness of **P-Companion** implies the benefits of diversified complementary modeling through item type transition instead of “product-level” modeling; (3) **P-Companion** provide out an innovative, scalable and end-to-end solution for a web-scale, high-quality and diversified product complementary recommender.

We also point out future directions and improvements. Particularly, instead of using type information extracted from product text features, we plan to adopt the product categorical ontology into our framework. Such hierarchy-aware categorical information will no doubt provide a more explainable and reliable way for neural-based recommendation systems. Another interesting direction is to leverage temporal customer purchase history information into **P-Companion** to enable personalized recommendation and help target the more desirable products.

CHAPTER 5

MEDTO: Medical Data to Ontology Matching Using Hybrid Graph Neural Networks

5.1 Introduction

In recent years, many medical ontologies have been created from various healthcare resources, semi-automatically or by human experts. Medical ontologies define, standardize and organize concepts in the medical domain, which provide valuable knowledge to support many healthcare applications, such as medical content browsing, clinical documentation, and evidence-based healthcare. Examples of medical ontologies include International Classification of Diseases (ICD), Unified Medical Language System (UMLS), and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT). Ontologies are widely used in data integration [DS13] and query federation to provide standard semantics across multiple systems. They are also used to enhance answers for medical databases using techniques known as Ontology-Based Data Access (OBDA) [XCK18].

Much of the literature in OBDA is devoted to the study of query answering under the assumptions that the ontology is available and the mappings between the database and the standard ontology have already been provided. However, we observe that these assumptions do not necessarily hold in real-life medical applications. A conversational system [QLM20] that we built for the medical database of IBM Micromedex[®], used by medical experts (e.g., doctors, nurses, pharmacists), revealed that the database was not designed with a target ontology and there was no mapping between the tables of the database and any known medical ontologies like UMLS or SNOMED CT. We also observe a similar issue on public medical

datasets such as MIMIC-III [JPS16], in which some tables and the associated columns are named with abbreviations or colloquial terms. These two use cases show the need for an end-to-end system that maps the medical database to standard medical ontologies, such that the downstream applications (e.g., OBDA) can benefit from the standard terminologies and vocabularies, discover additional relationships, and answer semantically rich queries.

There have been many efforts devoted to ontology matching [JG11, FPS13, KKK18], to find a mapping between two given ontologies. Ontology matching is only part of the problem we are trying to address in this work, namely *data to ontology matching*. When there is no ontology associated with a given database, we need to create an ontology describing the data as a first step, before we can apply ontology matching.

Most of the ontology matching work, such as LogMap [JG11] and AML [FPS13], rely on logical reasoning and rule-based methods to extract various sophisticated features from the ontologies. These terminological and structural features are then used to compute ontological concept similarities that drive the ontology matching. However, these features in one ontology often do not transfer in others. Consequently, the accuracy and robustness of ontology matching based on different features vary greatly with different medical ontologies to be matched [KKK18]. Worse yet, these solutions assume that the given ontologies are carefully crafted, which often fall short of the requirements for data to ontology matching.

Recently, graph representation learning [KW17, HYL17a] has emerged as an effective approach to learn vector representations for graph-structured data. The representation of a node is learned by recursively aggregating the representations of its neighboring nodes. Several studies [WLL18, WLF19, SWH20] have exploited graph neural networks (GNNs) for embedding-based entity alignment as similar entities usually have similar neighborhoods in knowledge graphs (KGs). Although existing GNN-based methods have achieved promising results on entity alignment in KGs, they are still facing three critical challenges when applied to data to ontology matching.

First, data to ontology matching often suffers from a cold-start problem, where a semantically rich ontology capturing a given medical database does not exist. One can generate

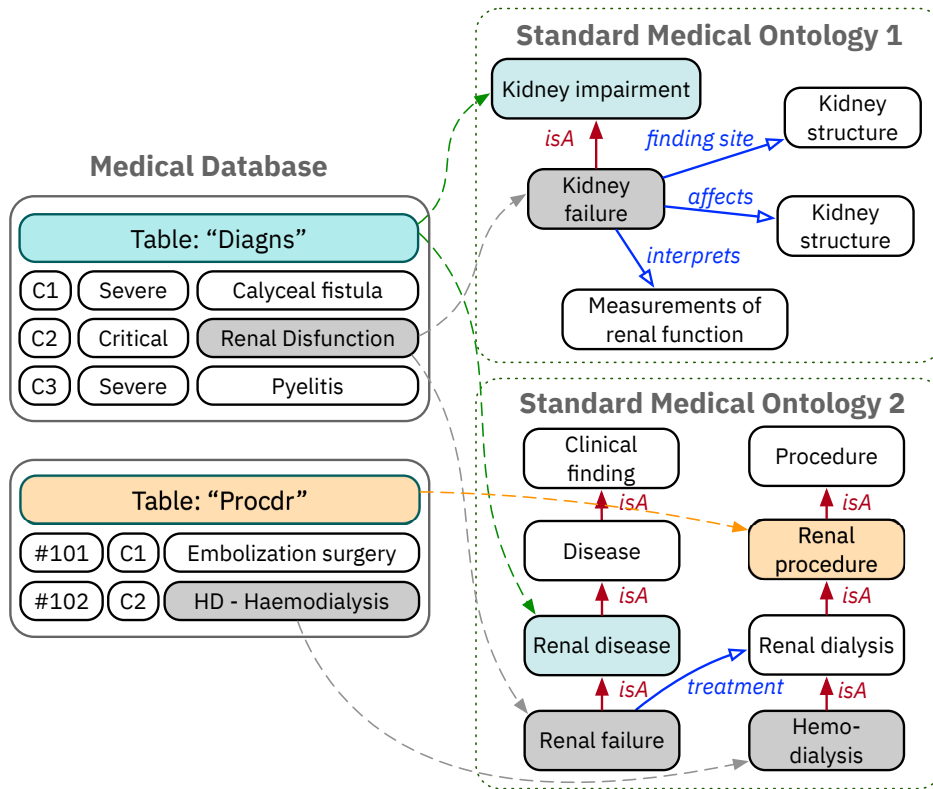


Figure 5.1: Example of data to ontology matching.

an ontology from the relational database [LOQ18] only using its metadata. However, we argue that if we enrich the ontology by using instance-level information from the database, and incorporate a richer set of semantic relationships, the derived ontology can be matched to the standard ontology with higher precision. To overcome this cold-start problem, a bootstrapping process is necessary.

Second, one distinct characteristic of medical ontologies, compared to the open-domain knowledge graphs like DBpedia [LIJ15] and YAGO [SKW07], is their deep domain specialization. These ontologies often have rich hierarchical top-down structures, which systematically organize medical concepts into categories and subcategories of different levels from general to specific. Figure 5.1 shows two snippets of medical ontologies. The hierarchical (through “isA” relations) neighborhood of “*kidney failure*” is very different from other types of relations. Capturing such hierarchical structures separately would help identify matching concepts and improve the accuracy of ontology matching.

Third, standard medical ontologies often are non-isomorphic in the local neighborhood structures of a concept from the one of a derived ontology. The rich and complex vocabularies, abundant sources of domain knowledge, and different modeling views all contribute to such non-isomorphism greatly. Fortunately, many medical ontologies have top-level concepts provided by domain experts, and such concepts provide a global context for matching concepts. In Figure 5.1, the concept “*clinical finding*” is the top-level category of “*renal failure*”. This helps us differentiate “*renal failure*” from other concepts such as “*renal dialysis*”, which belongs to the “*procedure*” category. Motivated by the fact that the semantically related latent information can appear in these top-level concepts, the aggregated neighborhood of a concept should include not only its local neighbors, but also the concepts with its global information.

To cope with these challenges, we propose a medical data to ontology matching (MEDTO) framework based on graph representation learning. The underlying idea is to first create and enrich a source ontology from the given medical database, and then embed both enriched and standard medical ontologies into two representations (i.e., hierarchical and non-hierarchical views) that are complementary to each other. Both representations are jointly optimized to improve the ontology matching capabilities. Our contributions are listed as follows:

- We propose an end-to-end framework MEDTO for data to ontology matching. MEDTO first bootstraps an ontology, based on a given medical database, and then learns and unifies hierarchical and non-hierarchical representations of two ontologies for matching.
- We design a lightweight yet effective method to create and enrich an ontology from the metadata of a medical database with rich semantic information from its instance data.
- We employ hyperbolic graph convolution layers to encode the parent and child concepts of each concept in the hyperbolic space, capturing the hierarchical characteristics in an ontology.
- To enrich the features of each concept, we introduce heterogeneous graph layers to incorporate both the local structure and the global context into concept embeddings.
- Our experiments on matching two real-world medical datasets to SNOMED CT show

that MEDTO significantly outperforms the state-of-the-art methods. We also evaluate MEDTO on a benchmark from the Ontology Alignment Evaluation Initiative (OAEI), showing that MEDTO consistently achieves state-of-the-art results.

5.2 Preliminaries and System Overview

5.2.1 Graph Neural Networks

Graph neural networks (GNNs) are deep learning based methods that operate on graph-structured data. It has been shown that GNNs are effective for various applications, such as node classification, link prediction and community detection. A generalized framework of GNNs [CAP20] consists of a graph encoder and a graph decoder, taking as input an adjacency matrix A , as well as optional node and edge features $X = \{X_N, X_E\}$. A typical graph encoder parameterized by Θ_{enc} combines the graph structure with node and edge features to produce node embedding matrix as:

$$Z = \text{ENC}(A, X, \Theta_{\text{enc}}). \quad (5.1)$$

The graph encoder uses the graph structure to propagate and aggregate information across nodes and learn embeddings that encode local structural information. A graph decoder is often used to compute similarity scores for all node pairs for downstream tasks on node, edge, or graph level.

Depending on the graph properties, a wide variety of GNNs have been developed. Representative examples include message-passing R-GCN [SKB18] and metapath-based HAN [WJS19] for heterogeneous graphs, non-Euclidean hyperbolic GCN [CYR19] for hierarchical graphs, and EvolveGCN [PDC20] for dynamic graphs. More details can be found in Section 5.6.

5.2.2 Problem Formulation

Definition 5.2.1. A medical database \mathcal{D} is represented by a relational schema \mathcal{S} and its instance \mathcal{I} . A schema is a finite collection of relation symbols. Each relation symbol has a specified arity, which intuitively corresponds to column names. An instance \mathcal{I} over \mathcal{S} is a collection of relations whose arities match those of the relation symbols in \mathcal{S} .

Definition 5.2.2. A medical ontology is represented as $\mathcal{O} = (\mathcal{C}, \mathcal{R}, \mathcal{T})$, where \mathcal{C} is the set of concepts, \mathcal{R} is the set of relations, and $\mathcal{T} = \mathcal{C} \times \mathcal{R} \times \mathcal{C}$ is the set of triplets.

Problem definition. Given a medical database \mathcal{D} and a standard medical ontology \mathcal{O} , the **data to ontology matching problem** is to find matches \mathcal{M} that map the schema \mathcal{S} of \mathcal{D} to \mathcal{O} , such that $\{(i, j) \in \mathcal{S} \times \mathcal{O} \mid i \equiv j\}$.

Note that a single standard medical ontology may only partially match with a medical database. In this case, multiple medical ontologies can be used to match against the given database in sequence. In essence, the challenges of matching data to ontology remain due to the semantically poor schema of the medical database and the complex structure of the medical ontology. Hence, we need to design an end-to-end system addressing these challenges.

5.2.3 System Overview

As depicted in Figure 5.2, we propose a framework, MEDTO, which consists of two phases: **data to ontology bootstrapping** and **ontology to ontology matching**. Given a medical database, the data to ontology bootstrapping phase first derives an ontology from its schema and data instances. It also bootstraps seed matches between the derived and standard ontologies by labeling highly confident matches and adding them into training data. The ontology to ontology matching phase takes as input the derived ontology, the standard ontology, as well as the seed matches (either provided or bootstrapped). Structures of both ontologies are captured via graph neural networks (GNNs) for structural representation learning. Moreover, the lexical semantics of the concepts in both ontologies is employed, providing complementary signals for ontology matching.

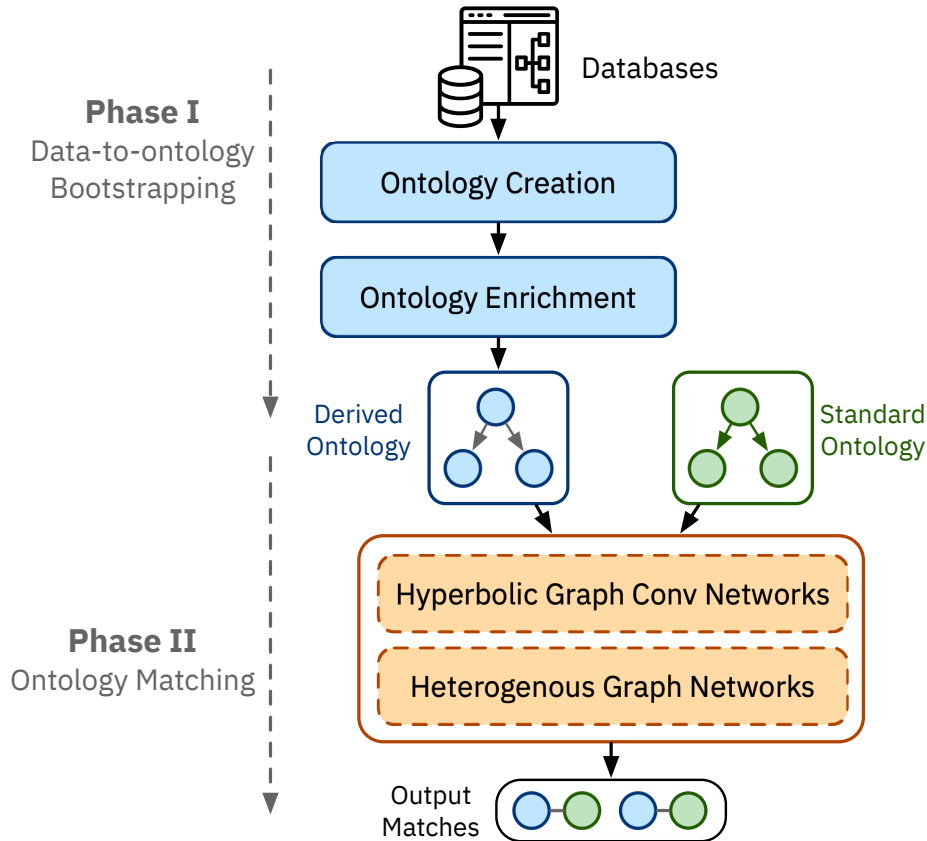


Figure 5.2: MEDTO system architecture.

5.3 Ontology Bootstrapping from Medical Database

In this section, we examine the ontology bootstrapping problem. Specifically, we first address the “cold-start problem”, i.e., the task of creating an ontology from a medical database. Then, we describe our novel concept augmentation and neighborhood augmentation strategies to enrich the derived ontology.

5.3.1 Ontology Creation

To infer an ontology (i.e., concepts and their relationships relevant to the domain) from a relational database, we leverage a variety of information from both database schema and data instances.

Concepts and properties. We map each table in a medical database to a concept and

represent columns in each table as data properties of that concept. Note that not all columns are selected, as they may not be semantically meaningful. Specifically, primary and foreign keys are not included, because they are designed for uniquely identifying each row in the table. Moreover, columns of non-string types (e.g., numeric, date, etc.) are not chosen either, since most standard medical ontologies only contain concepts expressed in strings.

Relation inference. Relation inference is non-trivial as it depends on the primary key and foreign key interactions, and quite often these keys are not specified in the databases, especially when the database is created from raw medical literature. Therefore, we follow the approach suggested in [LOQ18], which enables the inference of functional relations as well as concept hierarchies (i.e., isA relation).

In brief, we first identify primary and foreign keys by leveraging data statistics, such as distinct values. If the number of distinct values of the column and the total row count in the table are identical, we assert a primary key constraint. Similarly, for foreign keys, we check if the rows in the join of the two tables based on the selected columns are equal to the total rows of the referring table. Furthermore, we consider tables with exactly two columns, both acting as foreign keys to different tables in the schema, as intermediate tables. For every non-intermediate table R , we generate a functional relation that connects the concept C generated from R to another concept C' generated from the table R' , if one of R 's column is a foreign key referring to R' . If there is a table R_1 with a single column, which is a foreign key referring to a table R_2 , we consider the concept C_1 generated from R_1 as subsumed by the concept C_2 generated from R_2 . In this case, we assert an isA relation between two concepts corresponding to these two tables. Finally, the resulting ontology \mathcal{O}_1 is stored in OWL2 format.

5.3.2 Ontology Enrichment

Although the created ontologies capture schema-level details of the underlying data, they are far less semantically rich than the standard ontologies created by experts. To alleviate this issue, we introduce two effective augmentation heuristics to enrich the derived ontology

\mathcal{O}_1 from the medical database.

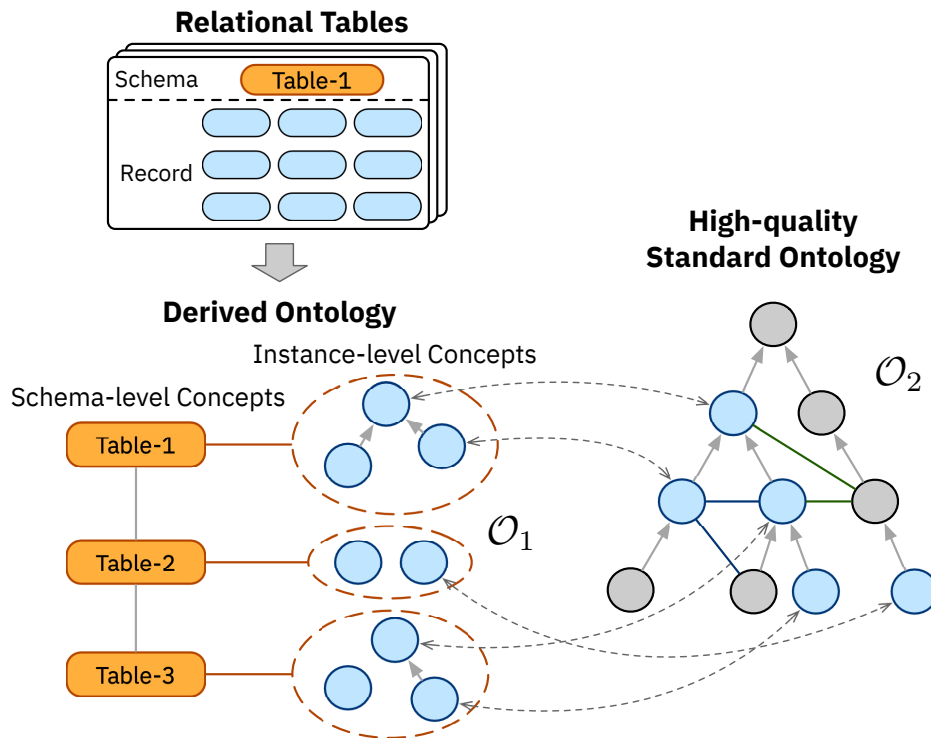


Figure 5.3: MEDTO ontology enrichment.

Concept augmentation. For each distinct value¹ in relational tables, we add an instance-level concept in \mathcal{O}_1 , and connect these new concepts to the existing schema-level ones via a new relationship “*instance of*”. The advantages of concept augmentation are twofold. First, it greatly enriches \mathcal{O}_1 with the available information from the relational database. Second, it enables us to bootstrap the seed concept matching between two ontologies using exact string matching algorithms. Other approximate string matching algorithms (e.g., edit-distance based or embedding based) or ML-based methods [SHZ18] can be plugged in as well, depending on the accuracy requirement.

Neighborhood augmentation. We also add edges among the pre-aligned seed concepts in \mathcal{O}_1 . Specifically, if two concepts i and j of \mathcal{O}_2 have an edge, while their counterparts i' and j' in \mathcal{O}_1 do not, we add an edge between i' and j' . The goal is to fill the semantic gap

¹If the number of distinct values is greater than a threshold, we use sampling to avoid exploding the ontology. We omit the details due to space constraints.

between \mathcal{O}_1 and \mathcal{O}_2 by adding the missing structural information.

With the augmented ontology, MEDTO can effectively learn the ontology representation and align it with \mathcal{O}_2 . To match a schema-level concept in \mathcal{O}_1 with the ones in \mathcal{O}_2 , we employ graph pooling to aggregate the embeddings of instance-level concepts that belong to the schema-level concept. Different graph pooling methods [HYL17a, YYM18] have been investigated for different scenarios. We find that the element-wise *mean*-pooling is sufficient to capture different information across the neighborhood set.

Finally, we feed both the enriched \mathcal{O}_1 and a standard ontology \mathcal{O}_2 into our novel graph neural network MEDTO to find the matches between them (Figure 5.3).

5.4 Ontology Matching

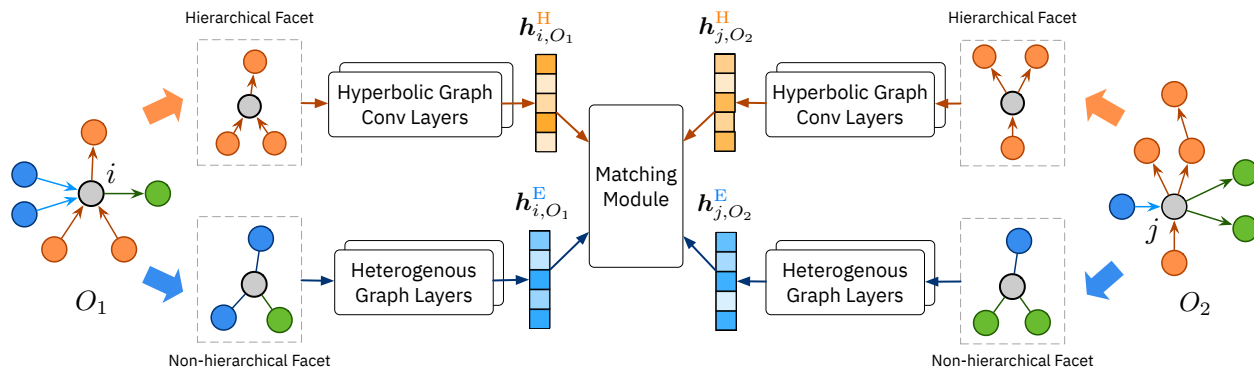


Figure 5.4: Details of MEDTO matching module. Both \mathcal{O}_1 and \mathcal{O}_2 are split into the hierarchical and non-hierarchical facets, which are fed into a hyperbolic graph layer and a heterogeneous graph layer respectively. The matching module minimizes the contrastive matching loss to let the representations of matching concepts have a very small distance while those of unmatched concepts have a large distance.

5.4.1 Input Embeddings of Medical Concepts

The concept names in a medical ontology consist of sequences of words. One can leverage deep learning based embedding methods such as BERT [DCL19] or ELMo [PNI18] to produce d_{in} -dimensional word embeddings for each concept. In this work, as the starting point, we

choose BioBERT [LYK20], a high-quality medical language model pre-trained on PubMed abstracts and clinical notes (MIMIC-III). The resulting input embedding is used as the initial state ($h^{0,E}$) of each concept, where E indicates that the embeddings are in a Euclidean space.

5.4.2 Hyperbolic Graph Convolution Layer

Conventional GNNs embed nodes into Euclidean space, which has been shown to incur a large distortion with hierarchical structures [NK17]. Hence, we use a hyperbolic embedding space, since it is amenable for learning concept hierarchies. Compared to Euclidean spaces, hyperbolic spaces better capture the hierarchical characteristic of ontologies. In this work, we adopt a specific model, hyperbolic graph convolutional neural network (HGNN) [CYR19], which leverages both the expressiveness of GNNs and hyperbolic geometry to learn node representations for graphs with hierarchical structures.

Hyperbolic graph convolution layer first establishes mapping between tangent (Euclidean) and hyperbolic spaces by exponential and logarithmic maps. We use the exponential map to project the node embeddings from a Euclidean space to a hyperbolic space, and logarithmic map reverses the map back to the Euclidean space. Hence, the initial embedding $h_i^{0,E}$ of node i to $h_i^{0,H}$ is:

$$\mathbf{h}_i^{0,H} = \exp_o^K \left(0, \mathbf{h}_i^{0,E} \right), \quad (5.2)$$

where K determines the constant negative curvature $-1/K$ ($K > 0$) and o denotes the origin in the hyperbolic space. For hyperbolic feature transformation from one layer to the next layer, we follow the definition below:

$$\mathbf{h}_i^{l,H} = \left(\mathbf{W}^l \otimes^{K_{l-1}} \mathbf{h}_i^{l-1,H} \right) \oplus^{K_{l-1}} \mathbf{b}^l \quad (5.3)$$

where \otimes and \oplus are hyperboloid matrix multiplication and addition, respectively, as defined in [CYR19].

Similar to GCN, our hyperbolic graph convolution layer aggregates features from a node’s local neighborhood. Since there is no notion of vector space structure in a hyperbolic space,

we have to map embeddings to the tangent space, perform the aggregation in the tangent space, and then map the aggregated embeddings back to the hyperbolic space. Furthermore, we utilize an attention mechanism to learn the importance of each neighboring node and aggregate neighbors' embeddings according to their importance. Given hyperbolic embeddings (h_i^H, h_j^H) , the attention weight w_{ij} is:

$$w_{ij} = \text{SOFTMAX} \left(\text{MLP} \left(\log_o^K (\mathbf{h}_i^H) \parallel \log_o^K (\mathbf{h}_j^H) \right) \right), \quad (5.4)$$

and the hyperbolic attention-based aggregation is:

$$\text{AGG}^K(\mathbf{h}^H)_i = \exp_{\mathbf{h}_i^H}^{K_i} \left(\sum_{j \in \mathcal{N}(i)} w_{ij} \log_{\mathbf{h}_i^H}^K (\mathbf{h}_j^H) \right), \quad (5.5)$$

where \parallel is a concatenation operation, and $\mathcal{N}(i) = \{j : (i, j) \in \mathcal{R}_{\text{isA}}\}$ denotes a set of parents of concepts $i \in \mathcal{C}$. Finally, we use a non-linear activation function to learn non-linear transformations by first applying the Euclidean non-linear activation in the tangent space and then mapping back to the hyperbolic space:

$$\sigma^{\oplus K_{l-1}, K_l}(\mathbf{h}^H) = \exp_o^{K_l} \left(\sigma \left(\log_o^{K_{l-1}} (\mathbf{h}^H) \right) \right). \quad (5.6)$$

The l -th layer of a hyperbolic graph convolution layer is:

$$\mathbf{h}_i^{l,H} = \sigma^{\oplus K_{l-1}, K_l} \left(\text{AGG}^{K_{l-1}} (\mathbf{h}^{l,H})_i \right), \quad (5.7)$$

where $-1/K_{l-1}$ and $-1/K_l$ are the hyperbolic curvatures at the $(l-1)$ -th and l -th layer, respectively. The hyperbolic embeddings at the last layer can be used to predict the concept similarity. We use the following sigmoid function [CYR19] to compute probability scores for edges:

$$\mathcal{L}^H = p((c_i, c_j) \in \mathcal{C}) = \left\{ \exp \left[\frac{1}{t} \left(d^K (\mathbf{h}_i^H, \mathbf{h}_j^H)^2 - r \right) \right] + 1 \right\}^{-1}, \quad (5.8)$$

where $d^K(\cdot, \cdot)$ is the hyperbolic distance and r and t are hyper-parameters.

5.4.3 Heterogeneous Graph Module

To capture the non-hierarchical structure in an ontology, conventional GNNs such as R-GCN [SKB18] can be applied, as it models multi-relational graphs. Specifically, R-GCN distinguishes different neighbors with relation-specific weight matrices. In the l -th convolutional layer, each representation vector is updated by accumulating the vectors of neighboring nodes through a normalized sum. Formally, the l -th layer of R-GCN is:

$$\mathbf{h}_i^{l,E} = \sigma \left(\mathbf{W}_0^l \mathbf{h}_i^{l-1,E} + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^l \mathbf{h}_j^{l-1,E} \right), \quad (5.9)$$

where W_0^l is the weight matrix for the node itself and W_r^l is used specifically for the neighbors having relation r , i.e., \mathcal{N}_i^r , \mathcal{R} is the relation set and $c_{i,r}$ is for normalization.

One limitation of this approach is that it focuses only on the local context of a concept and ignores the position of the concept within the broader context of the entire ontology. As described in Section 5.1, the top-level concepts in an ontology often provide additional semantic information which can influence how the final embeddings are aggregated. In Figure 5.5, the local context of “renal failure” includes two concepts, “measurements of renal function” and “kidney structure”, connecting to “renal failure”. In addition to the local context, we also incorporate the “global” context described by the top-level concepts, such as “clinical finding”, “body structure”, and “procedure”.

The key idea is to incorporate a set of “global” contexts and enrich each node’s feature with its corresponding global embeddings. We denote a node i ’s global embedding at l -th layer as $\mathbf{g}_i^{l,E}$. We replace the node feature $\mathbf{h}_i^{l-1,E}$ with its enriched version $\mathbf{h}_i^{l-1,E} \parallel \mathbf{g}_i^{l-1,E}$ and similarly replace each node feature of its neighbors $\mathbf{h}_j^{l-1,E}$ with concatenated $\mathbf{h}_j^{l-1,E} \parallel \mathbf{g}_j^{l-1,E}$ in Eq. 5.9. Note that in a medical ontology, a concept may belong to multiple top-level concepts. In this case, we take an element wise mean of all global embeddings to fully capture the global context. Such combined embeddings help us to learn better representations from more neighborhood information.

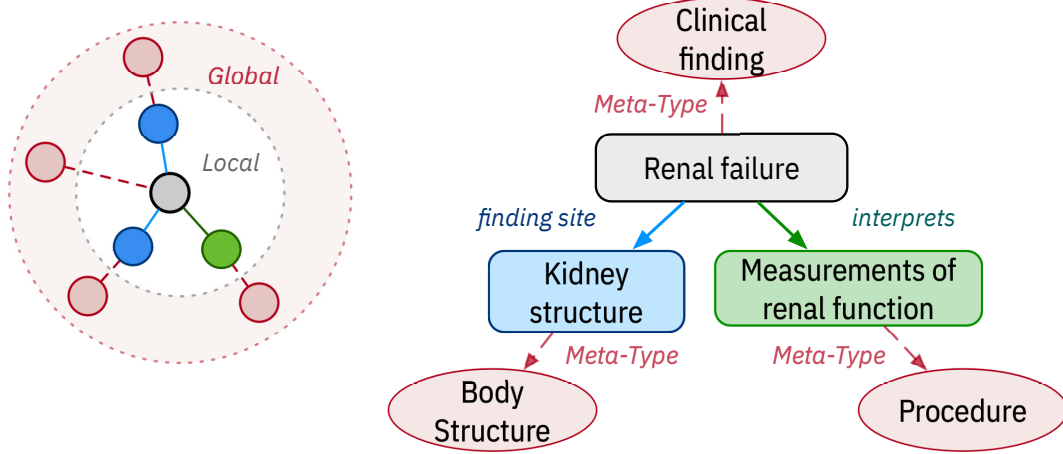


Figure 5.5: Local and global contexts of “renal failure”.

Following the convention, we optimize for cross-entropy loss to push the model to score observable edges higher than the negative ones:

$$\mathcal{L}^{\mathcal{E}} = \sum_{r \in \mathcal{R}} \sum_{i, j \in \mathcal{C}} w_{ij}^r \log \frac{\exp((\mathbf{h}_i^E)^T A_r \mathbf{h}_j^E)}{\sum_{i' \in \mathcal{C}} \exp((\mathbf{h}_{i'}^E)^T A_r \mathbf{h}_j^E)}, \quad (5.10)$$

where $w_{ij}^r = 1(i, j \in \mathcal{R}_r)$ and negative samples are generated by replacing i with a random node i' .

5.4.4 Matching Module

Based on the learned concept representations \mathbf{h}^H and \mathbf{h}^E from the hyperbolic convolution and the heterogeneous graph layers, we merge the two through concatenation to unify the representation of a concept \mathbf{h} . Then, the matching module $M(\cdot)$ takes pairs of concept embeddings from \mathcal{O}_1 and \mathcal{O}_2 and outputs the prediction score. We use the straightforward multi-layer perceptron (MLP) with one hidden layer, defined as follows:

$$M(\mathbf{h}_i^U, \mathbf{h}_j^U) = \sigma(\mathbf{W}_2 \cdot \gamma(\mathbf{W}_1(\mathbf{h}_i^U || \mathbf{h}_j^U) + \mathbf{b}_1) + \mathbf{b}_2), \quad (5.11)$$

where \mathbf{W}_1 , \mathbf{W}_2 , \mathbf{b}_1 , \mathbf{b}_2 are parameters, σ is the sigmoid function, and γ is the LeakyReLU activation function. A multi-head attention-based transformer encoder [VSP17] module can

also be used to replace the MLP.

We minimize the contrastive matching loss to let the embeddings of known matched concepts (positive) have a small distance while the unmatched (negative) pairs have a relatively large distance:

$$\mathcal{L}^M = \sum_{(i,j) \in \mathcal{M}^+} M(\mathbf{h}_i, \mathbf{h}_j) + \sum_{(i',j') \in \mathcal{M}^-} \omega [\lambda - M(\mathbf{h}_{i'}, \mathbf{h}_{j'})]_+, \quad (5.12)$$

where \mathcal{M}^+ denotes the seed matches between \mathcal{O}_1 and \mathcal{O}_2 , \mathcal{M}^- denotes a set of negative samples, λ is the margin value, ω is a balance hyper-parameter, and $[\cdot]_+ = \max(0, \cdot)$.

5.4.5 Training

Combining the hyperbolic graph convolution and heterogeneous graph models together with the matching module, MEDTO minimizes the final joint loss function:

$$\mathcal{L} = \mathcal{L}^M + \alpha_1 \cdot (\mathcal{L}_{\mathcal{O}_1}^H + \mathcal{L}_{\mathcal{O}_2}^H) + \alpha_2 \cdot (\mathcal{L}_{\mathcal{O}_1}^E + \mathcal{L}_{\mathcal{O}_2}^E), \quad (5.13)$$

where \mathcal{L}^M is the matching loss, $\mathcal{L}_{\mathcal{O}_1}^H$ ($\mathcal{L}_{\mathcal{O}_2}^H$) and $\mathcal{L}_{\mathcal{O}_1}^E$ ($\mathcal{L}_{\mathcal{O}_2}^E$) represent the losses of the hyperbolic graph convolution and heterogeneous graph models, respectively, and both α_1 and α_2 are positive hyper-parameters to control the trade-off among three loss components. We optimize all models with Adam [KB15a] optimizer.

5.5 Experiments

5.5.1 Datasets

We use the following datasets from the medical domain to evaluate the performance of our MEDTO framework.

MIMIC-III is a large database consisting of anonymized health-related data of over forty thousand patients who stayed in critical care units [JPS16]. It contains 21 tables in 3

aspects including patient tracking, ICU data, and hospital data.

MDX is a medical database of IBM Micromedex^{®2} that contains information in 59 tables about drugs, adverse effects, indications, findings, etc. It is manually curated from medical literature by editorial staff.

For standard medical ontologies, we choose the ones provided in the large BioMed track of OAEI³. This track consists of finding alignments between three ontologies: the Foundational Model of Anatomy Ontology (FMA), SNOMED CT, and the National Cancer Institute Thesaurus (NCI).

FMA is an ontology for biomedical informatics that represents a coherent body of explicit declarative knowledge about human anatomy [RM03]. It consists of 78,984 concepts and 78,985 isA relations.

NCI provides reference terminologies for clinical care, translational and basis research, and public information and administrative activities [CHS04], which consists of 56,907 concepts and 85,332 relations of 80 different types. 59,794 of them are isA relations.

SNOMED CT is a systematically organized collection of medical terms providing codes, terms, synonyms and definitions used in clinical reporting [Don06]. It contains 76,730 concepts and 109,896 relations, of which 105,563 are isA.

Seed matches are provided by OAEI and we split them into train, validation and test set as the positive samples. The negative samples are uniformly sampled by modifying one of the concepts in the positive sample pairs.

5.5.2 Compared Methods

To evaluate both phases of MEDTO, we compare our approach against a variety of methods in different categories. For MEDTO ontology bootstrapping phase, we choose the method introduced in ATHENA [LOQ18, JSM18] as the baseline, which only utilizes the schema in-

²<https://www.ibm.com/products/micromedex-with-watson>

³<http://www.cs.ox.ac.uk/isg/projects/SEALS/oei/2020/>

formation from a given database. For MEDTO ontology matching phase, the baselines range from rule-based methods to recent embedding-based entity alignment models. Specifically, LogMap uses logic-based reasoning over the extracted features and casts the ontology matching as a satisfiability problem. AML performs ontology matching based on heuristic methods that rely on aggregation functions. We select MTransE [CTY17], GCN-Align [WLL18], and RDGCN [WLF19]⁴ from recent embedding-based entity alignment methods. For ablation study, we develop three variants of MEDTO, i.e., MEDTO (w/o HYP) that does not capture the hierarchical information in the hyperbolic space, MEDTO (w/o HET) that does not pay attention to both local and global position information, and the full model MEDTO.

5.5.3 Implementation Details

The following hyper-parameters are used in the experiments. Each training took 1000 epochs with a learning rate of 0.01. The embedding dimension d is set to 128 for all the comparative methods (if applicable). The dimension of input embeddings is $d_{\text{in}} = 768$. By default, we stack 2 hyperbolic graph convolution and 2 heterogeneous graph layers in MEDTO. For the hyperbolic graph convolution decoder, we set $r = 2.0$, $t = 1.0$ (Eq. 5.8) and apply trainable curvature. In the matching module, we set $\lambda = 1.0$ and $\omega = 0.1$ (Eq. 5.12). We set both balance hyper-parameters α_1 and α_2 to 1.0 in Eq. 5.13. We sample 10 negative samples for each pre-aligned concept pair. All the learnable parameters are initialized by the Xavier initialization [GB10]. Following the convention of OAEI and entity alignment, we report the precision, recall and F1 score to assess ontology matching performance. In addition, we also report MRR (mean reciprocal rank), higher scores indicating better performance.

5.5.4 Experimental Results

Main results. We evaluate MEDTO on both MIMIC-III and MDX. For MIMIC-III, our domain experts identified 15 matching concepts in SNOMED, among 21 tables. For MDX,

⁴OpenEA library: <https://github.com/nju-websoft/OpenEA>

19 out of 59 tables have their matches identified in SNOMED as well. Hence, we use these identified matches as our ground truth. Following convention, we report Hits@10 and Hits@30 results to assess ontology matching performance.

Table 5.1: Matching MIMIC-III and MDX to SNOMED CT.

Dataset	MIMIC-III \Leftrightarrow SNOMED		MDX \Leftrightarrow SNOMED	
Metric	Hits@10	Hits@30	Hits@10	Hits@30
AML	0.06 (1/15)	0.13 (2/15)	0.16 (3/19)	0.26 (5/19)
LogMap	0.20 (3/15)	0.20 (3/15)	0.21 (4/19)	0.37 (7/19)
MTransE	0.00 (0/15)	0.00 (0/15)	0.05 (1/19)	0.05 (1/19)
GCN-Align	0.20 (3/15)	0.33 (5/15)	0.32 (6/19)	0.42 (1/19)
RDGCN	0.27 (4/15)	0.40 (6/15)	0.32 (6/19)	0.58 (11/19)
MEDTO	0.47 (7/15)	0.60 (9/15)	0.42 (8/19)	0.79 (15/19)

As shown in Table 1, MEDTO substantially outperforms all baseline methods. For MIMIC-III, the best performing baseline, RDGCN, can only find 4 matches when Hits@10, whereas MEDTO finds 7 out 15 matches. The primary reason is the concept and neighborhood augmentation we used to enhance the initially derived MIMIC-III ontology. With instance-level concepts and hierarchical relationships among them, MEDTO can leverage semantic information to learn much better representations of the MIMIC-III ontology, resulting in the performance gain. We observe similar results on the MDX dataset; our MEDTO finds 8 out of 19 matches compared to 3-6 matches found by other baselines, achieving superior performance. Tables 2 and 3 show a subset of successful and failed cases from both datasets.

Table 5.2: MDX-to-SNOMED result analysis (Hits@30).

MDX Tables	AML	LogMap	RDGCN	MEDTO
AdverseEffect	✓	✓	✓	✓
Dosage	✓	✓	✓	✓
DrugFoodInteraction	✗	✓	✓	✓
ContraIndication	✗	✗	✓	✓
DoseAdjustment	✗	✗	✗	✓
DrugRoute	✗	✗	✗	✗

We observe two mistake patterns from MEDTO. The first type of mistakes is caused by ambiguous semantic information. For example, most instance-level concepts of “*chartevent*” are described by different timestamps, which do not contribute to the bootstrapping of seed

matches between MIMIC-III and SNOMED at all. In fact, MEDTO solely relies on the input embedding of “*chartevent*”, which is not sufficient to locate the correct match in SNOMED.

Table 5.3: MDX-to-SNOMED result analysis (Hits@30).

MDX Tables	AML	LogMap	RDGCN	MEDTO
AdverseEffect	✓	✓	✓	✓
Dosage	✓	✓	✓	✓
DrugFoodInteraction	✗	✓	✓	✓
ContraIndication	✗	✗	✓	✓
DoseAdjustment	✗	✗	✗	✓
DrugRoute	✗	✗	✗	✗

The second type of mistakes still results from instance-level concepts augmented in MIMIC-III. Even though MEDTO is able to leverage these concepts to bootstrap the seed matches, these matches do not locate around the provided ground truth matches in SNOMED. For example, most instance-level concepts of “*labevents*” find the matches (e.g., “*hemoglobin*” and “*cholesterol*”) under “*substance*” concept in SNOMED. Consequently, MEDTO learns an incorrect representation of “*labevents*” and mistakenly matches it to concepts similar to “*substance*” rather than “*laboratory test*”. We observe similar trends from MDX case as well.

Ontology bootstrapping results. As mentioned earlier, we evaluate the effectiveness of MEDTO ontology bootstrapping methods against ATHENA [LOQ18, JSM18], which only utilizes the schema information of a database to create an ontology. MEDTO matches 7 out of 15, and 8 out of 19 over MIMIC-III and MDX, respectively, when Hits@10, while ATHENA is only able to match 3 out of 15, and 4 out of 19. Even when Hits@30, ATHENA (4 out of 15 on MIMIC-III, and 5 out of 19 on MDX) is still beaten by MEDTO substantially. The results clearly show that MEDTO ontology bootstrapping method can produce a semantically richer ontology compared to the one generated by ATHENA. Having the enriched ontology, the ontology matching phase can subsequently identify more matching concepts from the standard ontology.

Ontology matching results. Table 4 summarizes the results of ontology matching on three pairs of ontologies from OAEI datasets. We observe that MEDTO outperforms the three representative baselines from entity alignment, with an average improvement of 4.7%

Table 5.4: Results of ontology matching among FMA, NCI and SNOMED on OAEI datasets.

Dataset	FMA-NCI			
Metrics	Precision	Recall	F1 score	MRR
AML	0.942	0.899	0.920	–
LogMap	0.916	0.895	0.905	–
MTransE	0.627	0.640	0.633	0.416
GCN-Align	0.813	0.783	0.798	0.561
RDGCN	0.855	0.843	0.849	0.761
MEDTO	0.944	0.874	0.908	0.783
MEDTO (w/o HYP)	0.867	0.775	0.818	0.724
MEDTO (w/o HET)	0.927	0.851	0.887	0.763

(a) Ontology matching between FMA and NCI.

Dataset	FMA-SNOMED			
Metrics	Precision	Recall	F1 score	MRR
AML	0.902	0.729	0.806	–
LogMap	0.791	0.850	0.819	–
MTransE	0.505	0.475	0.490	0.372
GCN-Align	0.763	0.729	0.746	0.526
RDGCN	0.824	0.752	0.786	0.683
MEDTO	0.871	0.762	0.813	0.690
MEDTO (w/o HYP)	0.787	0.653	0.714	0.540
MEDTO (w/o HET)	0.863	0.747	0.801	0.676

(b) Ontology matching between FMA and SNOMED.

Dataset	NCI-SNOMED			
Metrics	Precision	Recall	F1 score	MRR
AML	0.890	0.744	0.810	–
LogMap	0.897	0.732	0.805	–
MTransE	0.254	0.378	0.304	0.349
GCN-Align	0.745	0.775	0.760	0.467
RDGCN	0.852	0.782	0.816	0.679
MEDTO	0.901	0.802	0.849	0.704
MEDTO (w/o HYP)	0.835	0.759	0.795	0.595
MEDTO (w/o HET)	0.881	0.807	0.842	0.688

(c) Ontology matching between NCI and SNOMED.

on F1 score and 2.5% on MRR. This indicates that entity alignment methods, designated for general-purpose knowledge bases (e.g., Wikidata and DBpedia), are insufficient for matching domain-specific medical ontologies with hierarchical structures. MEDTO explicitly distinguishes and models the hierarchical information, from other local and global structural features, leading to better results on medical ontology matching.

Compared to the extensively developed rule-based approaches (AML/LogMap), MEDTO achieves competitive results across all three datasets. In particular, MEDTO outperforms both AML and LogMap on NCI-SNOMED matching. It is the most challenging one among the three matching tasks, since both NCI and SNOMED are more complex than FMA. We also find that AML and LogMap heavily rely on lexical features from a suite of sophisticated matchers. Deriving such features for a given ontology can be time-consuming. However, these features in one ontology often do not transfer in others. As shown in Tables 2 and 3, the accuracy of such approaches varies dramatically depending on the quality of the given ontologies.

Effectiveness of Medto heterogeneous graph layer and hyperbolic graph convolution layer. We compare the performance between the proposed MEDTO and its two variations, named MEDTO (w/o HYP) and MEDTO (w/o HET), which only use the heterogeneous graph layers and hyperbolic graph convolution layers, respectively. Results are also shown in Table 2. We observe that full model MEDTO consistently performs the best across three datasets, with an average increase of 2.3% in F-1 score. This is attributed to MEDTO’s unified representation, capturing the critical semantic and structural features from multiple facets. It is also interesting to see that MEDTO (w/o HET) outperforms MEDTO (w/o HYP), which indicates that hierarchical information in medical ontologies contains more representative and critical features of ontology matching. Our hyperbolic graph convolution module effectively encodes such information for the matching module.

Hyper-parameter sensitivity analysis. We first analyze the results of MEDTO with 1 to 4 hyperbolic graph convolution and heterogeneous graph layers on OAEI datasets. In Figure 5.6a, we observe the optimal number of layers is 2 (for FMA-NCI and SNOMED-NCI)

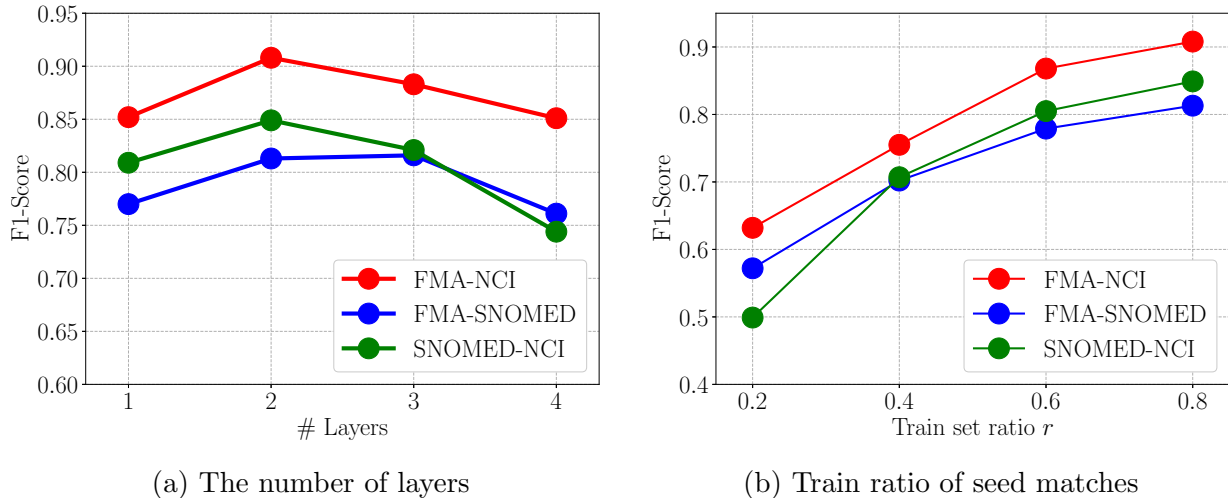


Figure 5.6: Sensitivity analysis.

or 3 (for FMA-SNOMED). When MEDTO uses more layers, its performance declines. The reason is that MEDTO indirectly captures more global contextual (i.e., top-level concepts) information by message propagation, and such global information would lead to more non-isomorphic neighborhoods.

Furthermore, we also aim to match ontologies with different numbers of seed matches. We use different proportions r of seed matches in OAEI datasets. As shown in Figure 5.6b, the MEDTO performs substantially better when r increases from 0.2 to 0.4, but the performance gain slows down as r increases from 0.6 to 0.8. This shows that MEDTO does not heavily rely on a large number of high-quality seed matches and provides decent matching results when the seed matches are limited.

5.6 Related Work

Graph representation learning. Recently graph representation learning has been intensively studied [HYL17a, SKB18] and shown effective for various tasks including node classification, link prediction and graph matching and multi-domain applications such as recommendation [HZL20] and even road traffic networks [DSW21]. Recently, graph attention networks [VCC18, WJS19, FZM20] have been introduced and allow each node to attend

over its various neighbors and uses attention to assign different weights to different nodes in a neighborhood. We refer interested readers to [DHW21] for more details. Recently, there has been research in extending GNNs to learn non-Euclidean embeddings and thus benefit from both the expressiveness of GNNs and hyperbolic geometry. Poincaré embeddings [NK17] learn embeddings of hierarchical graphs such as lexical databases (e.g., WordNet) in the Poincaré space. HGCN [CYR19] and HGNN [LNK19] apply graph convolutions in hyperbolic space by leveraging the Euclidean tangent space, which provides a first-order approximation of the hyperbolic manifold at a point. These methods lead to improvements on graphs with hierarchical structures.

Ontology matching. Traditional feature-based approaches have been investigated for ontology matching, including terminological-based features, structural-based features and employing external semantic thesauruses for discovering semantically similar entities. More specifically, LogMap [JG11] relies on lexical and structural indexes to enhance its scalability. AML [FPS13] also employs various sophisticated features and domain-specific thesauri to perform ontology matching. Feature-based methods mainly employ crafting features to achieve specific tasks. Unfortunately, these hand-crafted features will be limited for a given task and face the bottleneck of improvement. Representation learning has a recent impact on ontology matching. For instance, DeepAlignment [KKK18] is an unsupervised ontology matching system, which refines pre-trained word embeddings with the descriptions of entities, including synonyms and antonyms extracted from general lexical resources and information captured implicitly in ontologies. Similar to DeepAlignment, a framework is introduced for medical ontology alignment [KKS18], based on terminological embeddings. The retrofitted word vectors are learned from the domain knowledge encoded in ontologies and semantic lexicons.

Entity alignment. Similar to ontology matching, entity alignment seeks to find entities in different knowledge graphs (KGs) that refer to the same real-world object. With the recent success of graph representation learning, embedding-based entity alignment has emerged and attracted massive attention recently [SZH20]. GCNAlign [WLL18] leverages GCNs for

cross-lingual KG alignment. Entity alignments are discovered based on the distances between entities in the embedding space. RDGCN [WLF19] introduces dual relation graphs to capture complex relation information via attentive interaction between KGs. AliNet [SWH20] employs an attention mechanism to key distant neighbors to expand the overlap between entities neighborhood structures. It then controls the aggregation of the direct and distant neighborhood using a gating mechanism. We refer interested readers to the recent survey [SZH20] for more details on embedding-based entity alignment.

Data integration. Much effort has been made to towards data integration [DHI12, DS13], including schema alignment and data fusion. Schema mapping methods [CGH18] create a mediated (global) schema and identify the mappings between the mediated (global) schema and the local schemas of multiple databases to determine which attributes contain the same information. Hence, the main goal of data integration is to create the global schema so that multiple databases can be integrated and queried together. In data to ontology, on the other hand, there is only a single database, and the relations are mapped to concepts in a standard ontology to utilize standard vocabularies and enable semantically rich queries.

5.7 Conclusion

In this chapter, we propose an end-to-end framework MEDTO for medical data to ontology matching. MEDTO creates a semantically rich ontology from a given medical database and learns multiple facets of concepts in both enriched and standard ontologies. MEDTO encodes the hierarchical information of concepts in the hyperbolic space through hyperbolic graph convolution layers. We further capture both local and global structural information of concepts using heterogeneous graph layers. MEDTO incorporates the information from these layers and learns better concept representations for ontology matching. Our experiments on a variety of real-world medical databases and ontologies demonstrate the effectiveness of MEDTO. As future work, we plan to support different types of matching relations between two concepts (e.g., \subseteq , \supseteq , and disjoint) and to extend MEDTO to match data to multiple ontologies in a holistic manner.

CHAPTER 6

Multi-source Knowledge Graph Transfer

6.1 Introduction

Various large-scale information systems, such as knowledge bases (KBs), enterprise security systems, IoT computing systems and social networks [DCW17], exhibit comprehensive interactions and complex relationships among entities from multiple different and interrelated domains. For example, knowledge bases, such as DBpedia [ABK07], contain rich information of real-world entities (people, geographic locations, etc), normally from multiple domains and languages; and IoT systems contain thousands of mobile interrelated computing devices, mechanical and digital machines with various functions that constantly record surrounding physical environments and interact with each other. These systems can be formulated as heterogeneous graphs with nodes as system entities and edges as activities. Considering an enterprise security system as one example shown in Figure 6.1 (right), processes, internet sockets, and files can be treated as different types of nodes. Activities between entities, such as a process accessing a destination port or importing system libraries, are treated as edges in the graph. They can be utilized for many downstream tasks including identifying active entities or groups in social networks, inferring new knowledge in KBs and detecting abnormal behaviors [CZC16].

Due to the complex nature of real-world systems, it normally takes a long time, sometimes even months for newly-deployed information systems to construct a reliable graph “profile” to identify featured entities and activities. Therefore, there is a need to transfer and migrate knowledge (potential entities with corresponding high-confidence interactions) from other available sources provided by existing multiple well-developed systems. However, directly

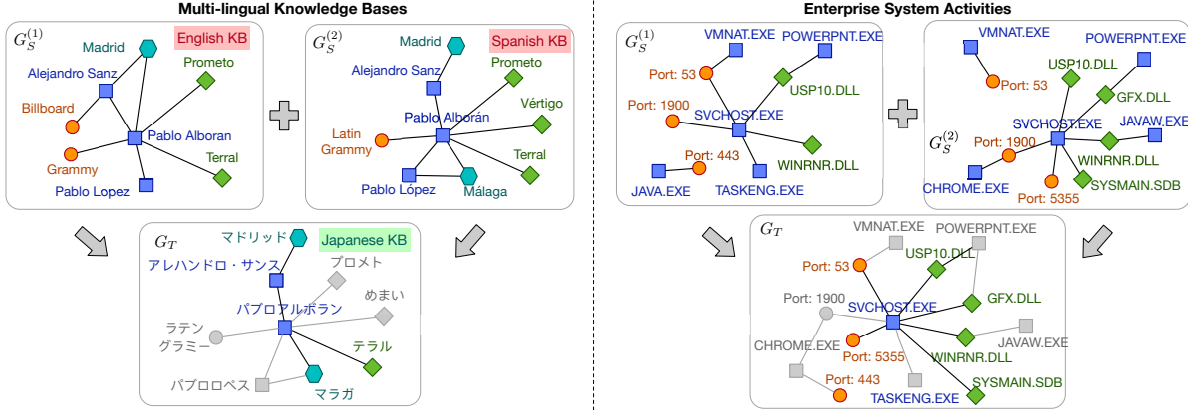


Figure 6.1: Two examples of multi-source graph transfer in knowledge bases (left) and enterprise systems (right). By leveraging the entities and relations from sources $G_{S^{(1)}}$ and $G_{S^{(2)}}$, we can estimate the target graph \hat{G}_T based on the current observation G_T . Grey nodes/links in \hat{G}_T denote new predictions from graph knowledge transfer. (Best viewed in color)

transferring existing nodes and links by copying is not reasonable and reliable enough since the source and target systems are not necessary for the exact same domains (e.g., transferring knowledge from existing departments to a new department in a corporation). It may transfer irrelevant or even incorrect entities and activities to the target graph. Existing research work [LCT18] mostly focuses on design learning frameworks for effective graph knowledge transfer between one source system and one target system and shows promising results on graph knowledge transfer. But in reality, it is quite common that multiple system sources are available. Simply using single-source graph knowledge transfer has its own limits: (1) the information from a single source is not sufficient in most cases; and (2) using only one source may lack generalization ability especially when the source and target are largely different, which leads to potential transfer failure. Learning graphs for newly-deployed systems through multi-source graphs will no doubt provide more comprehensive coverage of system entities and activities in multiple domains, and it will be more robust for downstream applications relying on learned target graph after selectively adapting knowledge from source graphs¹.

¹In this chapter, we use the *source graph* as the graph profiles for existing well-observed systems and *target graph* as the graph profile for new systems, which is relatively smaller than source graphs in graph size (e.g. number of nodes/edges). We assume that the number of source graphs is at least 2 and that of the target graph is 1.

Two application scenarios are shown in Fig. 6.1. In the case of multi-lingual KBs, low-resource KB (such as Japanese) can be enriched and improved with other KBs, and especially in the case of Pablo Alboran (Spanish pop singer), Spanish KBs can provide better and more accurate knowledge facts than others. Similarly in the example of enterprise systems, after the observation that the system has similar patterns of .dll connections of SVCHOST.EXE, a reasonable interpretation is that the target graph G_T will more likely grow more closely related patterns shown in source graphs.

However, the aforementioned selective multi-source transfer faces several challenges: (i) *How to represent multiple source graphs and target graphs effectively i.e. set up connections to leverage the graph knowledge in source graphs to the target graphs.* Not all sources are equally related to the target and it is required to differentiate multiple input source graphs in the transferring process, which is a difficult but important task to handle and will significantly affect the transfer performance. (ii) *How to handle potential conflicts on entities and interactions observed in multiple graphs.* The same interactions may be observed in some sources, but are not in others. In other words, there are potentially conflicting observations that cannot be easily tackled by simple transfer. In other words, if all sources are credited equally (for example, using one combined graph to include all the nodes and edges) and other methods that concatenate multiple graphs, one inductive bias is incorrectly assumed that nodes and/or edges are transferred and learned without selectivity and the approaches are subject to noise and misinformation on part of the sources.

To address the aforementioned tasks and corresponding challenges, we proposed a novel type of graph neural network designed for Multi-Source Graph Knowledge Transfer named **MSGT-GNN** which contains two model components: *Intra-Graph Encoder* and *Attention-based Cross-graph Transfer*. The high-level idea is that the knowledge transfer between the source and target graphs is done in a controllable manner where they are selectively learned. We employ self graph encoder model to a variety of state-of-the-art graph neural networks (GNNs) to obtain the node representations, that is, node embeddings learned from the node features itself and neighborhood in the context of the same source/target graph. On top of

the encoder model, the Cross-graph Transfer module adopts a novel attention mechanism based on both node level and graph level. This module can better learn the representations by attentively aggregating nodes in the broader context, which later applies in the graph decoder for link prediction. As a result, not only can we accelerate the process of graph enlargement to fast characterize the target graph, but we can also selectively and effectively leverage multiple sources in the information systems to estimate more reliable and accurate target graphs. Experimental results on target graph link prediction confirm that the effectiveness of MSGT-GNN and the performance of knowledge transfer significantly outperforms other state-of-the-art models including TINET.

6.2 Problem Statement

Given n multiple source domains $\mathcal{D}_S^{(i)}$ ($i = 1, 2, \dots, m$) and one target domain \mathcal{D}_T as input graphs have been on source domain for and these source graphs $G_S^{(i)}$ are stable already. Meanwhile, the system in \mathcal{D}_T is possibly newly deployed and therefore the target graph G_T incomplete and of relatively small size. Our goal is to transfer the graph knowledge (entity and edges) from $G_S^{(i)}$ ($i = 1, 2, \dots, n$) to G_T , and then help quickly enlarge and estimate an estimated complete graph \hat{G}_T to fit the domain of \mathcal{D}_T , which should be as close to the ground truth \bar{G}_T as possible. Note that in this work, we assume that alignments of the same entity among source and target graphs are well established, though such alignments are not fully feasible especially in knowledge bases. Under such formulation, we also point out that our proposed problem focuses on the graph enhancement from its incomplete status, different from temporal graph modeling where graphs are dynamically changed with multiple timestamps. Notations of all symbols used in this work are summarized in Table 6.1. Scalars, vectors and matrices are denoted with lowercase unbolded letters, lowercase bolded letters and uppercase bolded letters, if not explicitly specified.

We acknowledge that entity alignment may not be flawlessly given in many real-world applications and there are many existing research works lying on the direction of entity disambiguation, etc. As mentioned in Section 2, we point out that in this work we do

Table 6.1: Summary of important notations.

Notation	Description
$\mathcal{D}_S^{(i)}$	i -th source domain
\mathcal{D}_T	Target domain
$G_S^{(i)}$	The graph of the i -th source from $\mathcal{D}_S^{(i)}$
$\bar{G}_T, G_T, \hat{G}_T$	The ground-truth complete / incomplete / estimated complete graph of the target system from \mathcal{D}_T
$A_S^{(i)}, A_T$	The adjacency matrix of the i -th source graph $G_S^{(i)}$ / the target graph G_T
$\mathbf{Z}, \mathbf{Z}_S^{(i)}$	Embedding table for all N entities, or for $N_S^{(i)}$ entities from the i -th source graph (as output of graph encoders)
$\mathbf{h}_{S^{(m)}_i}^l, \mathbf{h}_{T_i}^l$	Embedding of the i -th node in the m -th source graph (or target graph) at the l -th layer of GNN (node embeddings, with node index)
$\mathbf{h}_{S^{(m)}}^l, \mathbf{h}_T^l$	Embedding of the m -th source graph (or target graph), at the l -th layer of GNN (graph embeddings, without node index)

not cover the scope of the entity alignments [TSD18, SZH20] (or entity resolution, entity conflation), which essentially predicts the correspondences of the same entity among different graphs. For example, in enterprise graphs, entities are generally identifiable with their IDs; in encyclopedic KGs, some labeled-property graphs are equipped with UID (universal identifier), which significantly reduces the alignment challenge. However, we believe such assumption can be relaxed, that is, MSGT-GNN can be further adapted to partially-given alignment or cross-graph alignment can be jointly learned, corrected, and/or enhanced, which is left as one direction of our future work.

6.3 Methodology

In this section, we formally propose **MSGT-GNN** to tackle multi-source graph knowledge transfer problem inspired by multi-task learning. As the model architecture of **MSGT-GNN** shown in Figure 6.2, it breaks down into two components: *Intra-graph encoder* and *Cross-Graph transfer*, which are explained in Section 6.3.1 and Section 6.3.2 respectively.

6.3.1 Intra-Graph Encoder

Generally, a graph encoder serves a function to represent nodes by their embeddings, from the original node features (categorical attributes, textual descriptions, etc), based on the graph features. Our proposed Intra-Graph Encoder, as the first component in MSGT-GNN, aims to learn the node features in the context of its own graph (source or target), i.e. the graph to which it originally belongs.

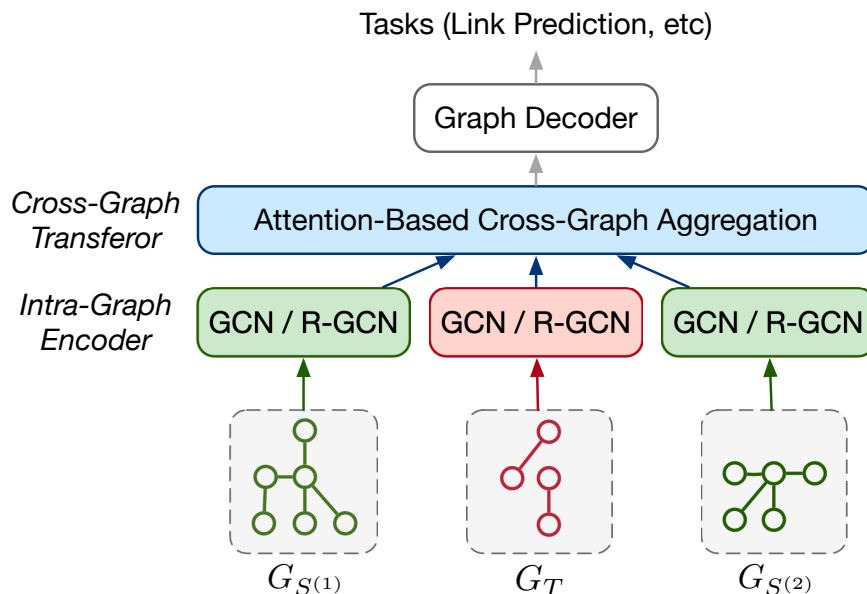


Figure 6.2: Model architecture overview for MSGT-GNN (two source graphs are shown). Node embeddings across multiple graphs are learned through two-module framework, i.e. *Intra-graph encoder*, which learns node embeddings of its own graph context from initial node features; and *Cross-Graph transferor*, which enables learning through multiple graphs and node embeddings are updated by its corresponding nodes in other source as well as the graph-level information.

As discussed in Section 6.5, graph neural networks (GNNs), deep learning based approaches that operate on graph-structured data, have recently shown effective for various applications such as node classification, link prediction and community detection. A generalized framework of GNNs consists of such a graph encoder, taking as input an adjacency matrix A , as well as original (optional) node features $X = \{X_N\}$. A typical graph encoder parameterized by Θ_{enc} combines the graph structure with node features to produce node embeddings as, $Z = \text{ENC}(A, X, \Theta_{\text{enc}})$, where Z is the learned comprehensive representation

from GNNs and is used for downstream tasks with designated graph decoders.

More specifically, in **MSGT-GNN**, for homogeneous graphs, we choose the Intra-Graph Encoder as standard GCN [KW17], which can be described as,

$$\mathbf{H}_i^{(l+1)} = \sigma \left(\hat{D}^{-\frac{1}{2}} \hat{A}_G \hat{D}^{-\frac{1}{2}} \mathbf{H}_i^{(l)} W^{(l)} \right), \quad (6.1)$$

where $\mathbf{H}_i^{(l)} \in \mathbb{R}^{n \times d}$ are embedding of after l -th GCN layers and $\hat{A}_G = A_G + I$ where I is the identity matrix, A_G is adjacency matrix of given graph G , \hat{D} is the diagonal node degree matrix of \hat{A} , as defined in [KW17]. Note that G can be either any source graph $G_{S^{(i)}}$ or target graph G_T . For multi-relational heterogeneous graphs such as knowledge graphs and enterprise systems, we adopt R-GCN [SKB18], which utilizes relation-wise weight matrix,

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\mathbf{W}_0^l \mathbf{h}_i^{(l)} + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^l \mathbf{h}_j^{(l)} \right), \quad (6.2)$$

where \mathbf{W}_0^l is the weight matrix for the node itself and \mathbf{W}_r^l is used specifically for the neighbors having relation r , i.e., \mathcal{N}_i^r , \mathcal{R} is the relation set and $c_{i,r}$ is for normalization. Similarly, R-GCN applies both in the source graphs and the target graph. In both cases, the number of GNN layers L is one hyperparameter².

6.3.2 Attention-based Cross-graph Transfer

The goal of our proposed *Cross-graph Transfer* is to provide a valid transfer mechanism in the entity embedding space for multi-source graphs. It is built on top of the Intra-Graph Encoder to enable the node embeddings selectively updated by the cross-graph “neighborhood” in both node level and graph level attention mechanism. Details of *Cross-graph Transfer* are shown in Figure 6.3.

To prepare for cross-graph transfer, one necessary module is Graph-level Aggregator, which takes the set of node representations and compute graph level representation, as

²In this work, the performance is relatively insensitive to L where we fix $L = 2$ for GNN modules including baselines

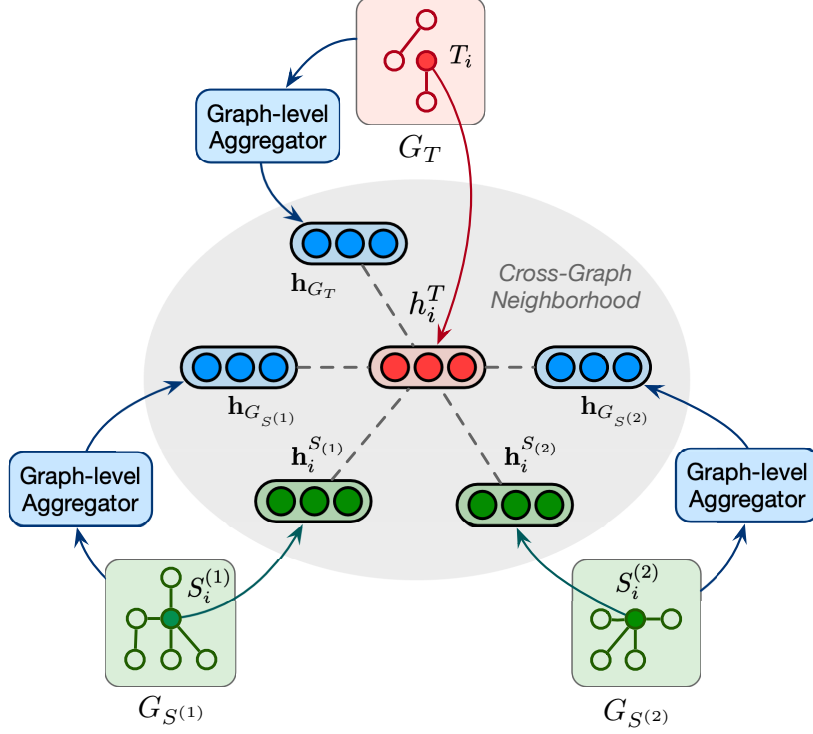


Figure 6.3: Details about Cross-Graph Transfer Layer operating on the Node T_i , updated by itself and its corresponding cross-graph neighbors (node-level embeddings), attentively learned from graph-level embeddings (Best viewed in color)

$\mathbf{h}_G = f_G(\{\mathbf{h}_i^G\})$ where $\mathbf{h}_G \in \mathbb{R}^d$, for both source and target graphs³. We use the MLP aggregator following the implementation in [LGD19]. The aggregation function operating on a node i of the target graph is defined as,

$$\mathbf{h}_{T_i}^{l+1} = \sigma \left(\mathbf{W}_0^l \mathbf{h}_{T_i}^l + \sum_m \alpha_m \mathbf{W}_n^l \mathbf{h}_{S(m)_i}^l \right), \quad (6.3)$$

where \mathbf{W}_0^l is the weight matrix for the node itself and \mathbf{W}_n^l is used specifically for the cross-graph neighbors (from the given alignments), of the l -th layer. $\mathbf{h}_{S(m)_i}^l$ denotes the l -th layer's hidden representation of node i in $G_{S(m)}$. α_m is attention weight computed over all m cross-graph neighbors. as,

$$\alpha_m = \text{softmax} \left(\left[\mathbf{h}_i^{S(m)}; \mathbf{h}_{G_{S(m)}} \right]^T \cdot \mathbf{W}_{\text{att}} \cdot \mathbf{h}_{T_i}^l \right), \quad (6.4)$$

³Theoretically the embedding dimension of graph-level representation can be different from that of the node-level. For simplicity, we choose both dimensions are the same, that is, $\dim(\mathbf{h}_G) = \dim(\mathbf{h}_{G_i}^l)$, where G refers to either source or target graph.

where $\mathbf{W}_{\text{att}} \in \mathbb{R}^{2d \times d}$ and $[\mathbf{h}_i^{S(m)}; \mathbf{h}_{G_S(m)}]$ is the concatenation of node-level cross-graph neighbor embedding and the graph-level embedding. By such cross-graph transfer, the node in one graph will be consequently updated and optimized attentively by nodes from other associated graphs. It is noteworthy to point out that our proposed MSGT-GNN does not explicitly differentiate source graphs and target graphs, which means the learned embeddings are not limited to make predictions over the target graph.

6.3.3 Graph Decoder

Graph Decoder and training objective The graph decoder use the learned representation from MSGT-GNN for link prediction during the inference stage. For homogeneous graph, we apply inner product to represent the edge plausibility, which is $\text{DEC}(\mathbf{Z}) = \mathbf{h}_i^T \mathbf{h}_j$ where $\mathbf{h}_i, \mathbf{h}_j \in \mathbf{Z}$ (\mathbf{h} is the learned embedding table for all nodes). For multi-relational graph, we apply DistMult score function [YYH15] to represent the edge plausibility, which is $\text{DEC}(\mathbf{Z}) = \mathbf{h}_i^T \mathbf{D} \mathbf{h}_j$ where $\mathbf{h}_i, \mathbf{h}_j \in \mathbf{Z}$ and \mathbf{D}_r is a diagonal matrix for relation r . Therefore, the training objective is,

$$\mathcal{L}_G(\mathbf{Z}_G) = (\mathbf{Z}_G \mathbf{D}_r \mathbf{Z}_G^T - A_G)^\theta + \Omega(\mathbf{Z}_G), \quad (6.5)$$

where $\theta = 2$ in practice and $\Omega(\mathbf{Z}_G, \mathbf{w}) = \lambda \|\mathbf{Z}_G\|_F$ is regularization term. $\mathbf{D}_r = I$ for homogeneous graph.

6.3.4 Training, Inference and Complexity

Joint training on source and target graphs Considering all the source and target graphs, MSGT-GNN minimizes the joint loss with meta-path similarity matrices for multiple graphs, $\mathcal{L} = \mu \sum_i \mathcal{L}_{S^{(i)}} + (1 - \mu) \mathcal{L}_T$, where $\mu \in (0, 1)$ is a hyperparameter that explicitly balances the importance of source and target graphs. We use the Adam [KB15a] to optimize the joint loss.

Inference During the inference stage, similar to other graph neural networks with downstream link prediction task, two steps of graph encoders (intra-graph and cross-graph) en-

codes pairs of nodes (from the target graph only for valid testing) into their representations through the trained GNN with the neighbor nodes (both inside its own graph and other sources) weighted by the graph-level representations. Later such embeddings are forwarded to graph decoder for link prediction which outputs plausibility scores of the given potential edges, as link prediction results.

Complexity Analysis For MSGT-GNN with the direct encoder, the overall runtime complexity is $\mathcal{O}(tnd|E|)$, which is linear to the size of total edges in multiple source graphs ($|E|$ is the total number of links in source/target graph). As for model parameter complexity, including all embeddings and transformation functions, the result is $\mathcal{O}(|V|d + nd^2)$ ($|V|$ is the total number of nodes in source/target graphs).

6.4 Experiments

6.4.1 Datasets

Three datasets on the knowledge bases, enterprise security and academic scholar community are used in the experiments. Data from a real-world enterprise system are collected from 145 machines from 4 departments (3 used as sources and 1 used as a target) in a period of 30 days, with a size of 3.45GB after integration and filtering. The entire enterprise security system contains both Windows and Linux machines and we consider they are disjoint graphs as datasets (named as **Windows and Linux Dataset**). Similar to the example in Figure 6.1, the entities (nodes) in all graphs are processes, internet sockets and libraries (mostly .dll files) and interactions (edges) between the process to file, process to process and process to internet sockets are observed as links in the dataset.

We also consider alternative datasets that are publicly available and from diverse domains are, (i) encyclopedia knowledge bases i.e. **DBpedia** [ABK07], extracted from five languages (en, es, de, fr, ja) of variant graph sizes and completeness; and (ii) **Aminer**, as one academic scholar community dataset [TSW09]⁴ from Aminer on five data mining/machine

⁴We use a subset of the co-author networks, which is available at <https://aminer.org/data#>

Table 6.2: Dataset statistics.

Dataset	Scholar	Enterprise		DBpedia
		Windows	Linux	
# Graphs	3	5	5	5
# Rel. Types	1	3	3	96
# Nodes	2.1k	10.7k	8.9k	12.5k
# Edges	9.0k	87.9k	62.5k	278.1k

learning related research communities in the past years. The nodes are authors and links are simply co-author relationships, which is essentially a homogeneous graph. More specifically dataset, we consider different languages as different domains in the context of **MSGT-GNN**, and given the graph size of these languages, we adopt two disjoint settings: $\{\text{en,fr,de}\} \rightarrow \text{ja}$ ⁵ and $\{\text{en,fr,de}\} \rightarrow \text{es}$. This results in a total of 5 datasets from 3 domains in our experiments. More details are listed in Table 6.2.

6.4.2 Baseline Methods

We compare our proposed model **MSGT-GNN** with the following baseline methods:

No Transfer (NT) directly uses the original observed incomplete target graph without any knowledge transfer, that is, $\hat{G}_T = G_T$.

Direct Union Transfer (DUT) directly combines all source graphs and the incomplete target graph, as prediction (“union” graph). That is, DUT outputs a union set on entities and links from all observed graphs without any selection, which means, $\hat{G}_T = G_T + \left(\bigcup_i G_S^{(i)}\right)$.

TINET applies the single graph knowledge transfer framework [LCT18]. To fit the multi-source setting, we choose three variations about TINET models: (i) to use the closest⁶ source graph as the transfer source, named **C-TINET**; (ii) to use the union graph as defined in DUT, as the single transfer source, named **U-TINET**; iii to use TINET iteratively on

Topic-coauthor.

⁵ $\{\text{en,fr,de}\} \rightarrow \text{es}$ means the source graphs are from DBpedia English, French and German KBs and the target is Spanish KB.

⁶Default similarity between the source and target graph is based on the Jaccard index.

Table 6.3: Results of KG triple completion. H@1 and H@10 denote *Hit@1* and *Hit@10* respectively. For each group of model variants with the same intra-view model, the best results are bold-faced. The overall best results on each dataset are underscored.

Dataset	Scholar
NT	0.526 ± 0.000
DT	0.398 ± 0.000
C-TINET	0.635 ± 0.009
U-TINET	0.618 ± 0.015
W-TINET	0.644 ± 0.017
O-TINET	0.622 ± 0.014
UT-GCN/RGCN	0.606 ± 0.025
UT-GAT/KGAT	0.635 ± 0.018
Intra-Only GCN/RGCN	0.597 ± 0.014
Intra-Only GAT/KGAT	0.624 ± 0.020
UDA-GCN	0.652 ± 0.017
MSGT-GNN	0.668 ± 0.016

Dataset	Enterprise: Windows	Enterprise: Linux
NT	0.664 ± 0.000	0.656 ± 0.000
DT	0.480 ± 0.000	0.578 ± 0.000
C-TINET	0.727 ± 0.008	0.759 ± 0.009
U-TINET	0.718 ± 0.012	0.733 ± 0.008
W-TINET	0.739 ± 0.011	0.772 ± 0.017
O-TINET	0.715 ± 0.012	0.740 ± 0.010
UT-GCN/RGCN	0.700 ± 0.030	0.722 ± 0.019
UT-GAT/KGAT	0.744 ± 0.023	0.750 ± 0.015
Intra-Only GCN/RGCN	0.745 ± 0.012	0.734 ± 0.014
Intra-Only GAT/KGAT	0.742 ± 0.018	0.738 ± 0.021
UDA-GCN	0.735 ± 0.013	0.727 ± 0.016
MSGT-GNN	0.776 ± 0.021	0.768 ± 0.018

Dataset	Encyclopedia:{en, fr, de}→ja	Encyclopedia:{en, fr, de}→es
NT	0.475 ± 0.000	0.545 ± 0.000
DT	0.299 ± 0.000	0.408 ± 0.000
C-TINET	0.596 ± 0.010	0.764 ± 0.013
U-TINET	0.617 ± 0.014	0.750 ± 0.012
W-TINET	0.645 ± 0.022	0.779 ± 0.018
O-TINET	0.620 ± 0.009	0.766 ± 0.011
UT-GCN/RGCN	0.576 ± 0.022	0.756 ± 0.026
UT-GAT/KGAT	0.559 ± 0.012	0.710 ± 0.014
Intra-Only GCN/RGCN	0.661 ± 0.015	0.739 ± 0.021
Intra-Only GAT/KGAT	0.656 ± 0.016	0.724 ± 0.016
UDA-GCN	0.610 ± 0.024	0.688 ± 0.022
MSGT-GNN	0.685 ± 0.018	0.801 ± 0.028

multiple sources, i.e. transferring one source once in an order, named **O-TINET**. Best performance is reported among all transfer orders.

W-TINET This method uses the weighted version of TINET for source and target graphs. Extending the single-source graph knowledge transfer model to multi-source, we adopt the same sub-model components (EEM, DCM) but adjust the objective function to be the sum of all source graphs.

Intra-Only GNN only uses *Intra-Graph Encoder* component in MSGT-GNN and discards the *Cross-Graph Transfer*. That is, standard GCN [KW17] is applied for homogeneous graphs and R-GCN [SKB18] is applied for multi-relational graphs which preceded the graph decoder. Alternatively, we also consider existing attention-based graph neural networks (applied on a single graph) i.e. GAT [VCC18]/KGAT [WHC19] as replacement of GCN/R-GCN. (Denoted as “Intra-Only GCN/RGCN” and “Intra-Only GAT/KGAT” respectively).

UT-GNN Similar to Intra-Only GNN, this method applies *Intra-Graph Encoder* component only on the “union graph” from the DUT method which forms one combined graph instead of multiple sources and target graphs. Two options (GCN/RGCN, GAT/KGAT) are still considered except the different graph inputs. (Denoted as “UT-GCN/RGCN” and “UT-GAT/KGAT” respectively)

UDA-GCN It develops a dual attention-based graph convolutional network component and domain adaptive learning module, which jointly exploits local and global consistency for feature aggregation to produce unified representation for nodes. We replace the decoder module⁷ for link prediction instead of node classification in the previous work [WPZ20].

6.4.3 Experiment Setup

Evaluation Protocol Similar to [LCT18], we adopt F1 score to evaluate the accuracy of the graph completion task on the target system instead of Hit@K or MRR score in knowledge

⁷Original code implementation: <https://github.com/GRAND-Lab/UDAGCN>

graph completion ⁸. In our experiment for multi-graph knowledge transfer, the main result is reported as the average and standard deviation of link prediction (edge) F1 score. As F1 score generally is the harmonic mean of precision and recall, we hereby define the *precision* and *recall* by comparing the estimated links between entities with the ground truth. The precision and recall are defined as: $Precision = N_C/N_E$ and $Recall = N_C/N_T$, where N_C is the number of correctly estimated links, N_E is the number of estimated links in total, and N_T is the number of the ground-truth links. For training, as mentioned in Section 6.2, we choose one incomplete target graph as the “new” system and complete source graphs from the rest as “old” systems and for training. In addition, we use $m = l/l_{full}$ as an index of “*graph maturity*”, which is defined as the observed number of edges (in training set) l of the target graph and the total number of edges l_{full} recorded in the ground truth target graph.

Hyperparameters In the experiment, we set $m = 0.4$ and $d = 128$ if not specified. The number of GCN/R-GCN layers in Intra-Graph Encoder is set as 2 and The number of Cross-Graph Transfer layers is set as 1. Default node embeddings are initialized by either node categorical features (scholar and enterprise dataset) or BERT sentence embeddings from entity descriptions (KB datasets). Hyperparameters are discussed in Section 6.4.5.

6.4.4 Results

Results on the target graph completion task are shown in Table 5.4. We observe that **MSGT-GNN** outperforms other baselines in terms of average F-1 score. Especially compared with non-transfer, **MSGT-GNN** achieves an average increase of 0.05 on F1 score among all datasets, which proves that **MSGT-GNN** transfers useful graph knowledge to the target. Also, **MSGT-GNN** outperforms all the TINET variants in the average F1 score especially on U-TINET and W-TINET which indicates that **MSGT-GNN** adopts a more effective strategy to use multi-

⁸We point out the thread of KG embedding in Section 6.5, including TransE and recent variants [WMW17]. The limitation of such methods is that they are transductive methods. This is generally not applicable to our inductive learning and its downstream link prediction. However, as for evaluation metrics, we follow the metrics adopted in previous work [LCT18] for target-adapted edge prediction instead of MRR or Hit score for a different triple completion task.

ple sources and learn better latent feature representations of entities with the process graph encoding and domain transferring. Since TINET follows a two-stage (entity selection and edge prediction), the performances significantly decrease when wrong or incomplete entity set is selected for subsequent link prediction. Unlike TINET and its variants, **MSGT-GNN** adopt end-to-end model architecture without explicit steps of entity/node selection. Comparing **MSGT-GNN** and standard GCN/R-GCN or GAT/KGAT, we also observe that **MSGT-GNN** achieves better link prediction performance with a relative gain of 4.9%, which shows the benefit of Cross-Graph Attention Transfer, which can better characterize node latent representations from actively and selectively aggregating useful information from the cross-graph neighborhood. It is noteworthy that NT directly uses the currently observed target graph (incomplete) as output; DT means the union set of all G_S and G_T without any selection. Typically DT includes much more noise and unwanted information into the target graph compared “beneficial section of transfer”, i.e., lots of links/edges are falsely predicted as positive. A similar observation is also reported in one of our baselines, TINET. Furthermore, we observe that GAT/KGAT variants almost have similar performance on the task (sometimes even worse). We hypothesize that the attention mechanism adopted by the original GAT/KGAT cannot best selectively learn the knowledge transfer in the cross-graph setting, although recent research shows that they outperform GCN/RGCN on the intra-graph node classification task. It is also noticed that UT-GNN generally performs worse than the Insta-Only setting which indicates that the union graph which equally combines the source graphs without selection has inductive biases which compromise the knowledge transfer in link prediction on the target graph.

6.4.5 Hyperparameters

In this section, we primarily investigate the sensitivity of target graph input maturity m , embedding dimension d and balance weight μ between the source and target graphs.

Graph maturity m We vary the target graph input by controlling the graph maturity m (let $m = \{0.2, 0.4, 0.6, 0.8, 1.0\}$). From Figure 6.4, we observe that, for both Windows

and DBpedia: $\{en,fr,en\} \rightarrow es$ graph, the performance of all models increases when the graph maturity m increases. As other approaches achieve F1 score of 1 when m gets close to 1, direct transfer only achieves around 0.60 as F-1 score, which seems not effective because all the irrelevant entities and links are adopted in the output target graph prediction. On the other hand, given the same level of graph maturity, **MSGT-GNN** achieves the best performance among all other methods on all datasets.

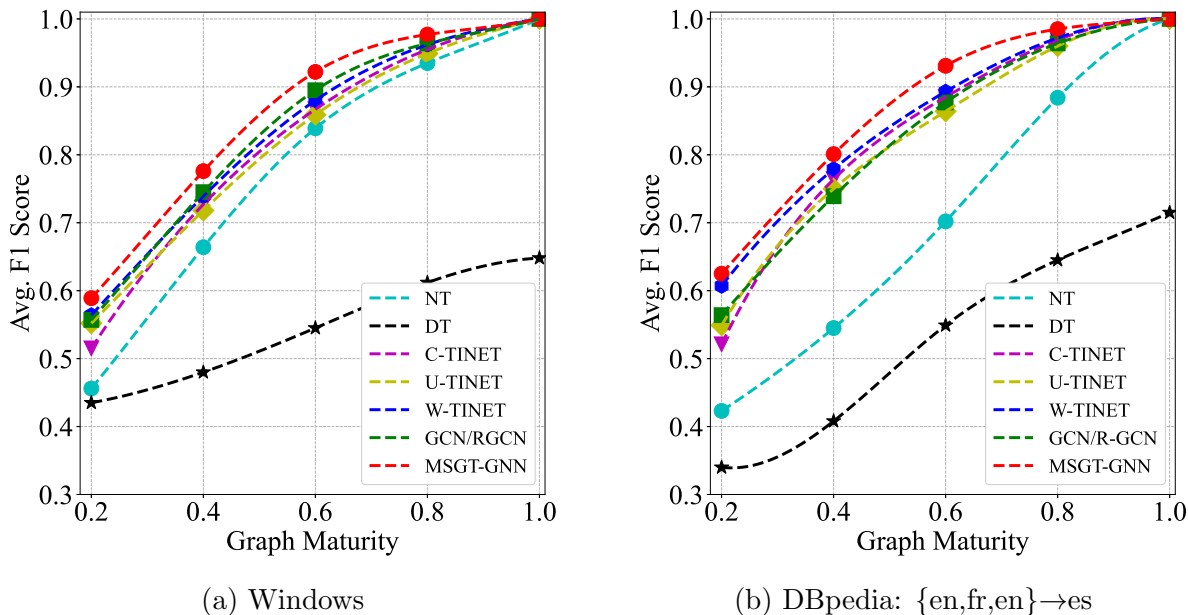


Figure 6.4: Performances with graph maturity. Most models achieve average F1 score close to 1 as the maturity of input observed target graph grows, while **MSGT-GNN** outperforms other baselines.

Dimensionality Dimensionality is a key hyperparameter that affects the quality of the obtained embeddings. Figure 6.5 shows the performance of **MSGT-GNN** on both Windows and DBpedia: $\{en,fr,en\} \rightarrow es$ dataset according to different embedding dimensions $d \in \{64, 128, 256, 512\}$ (both local and global embeddings) on the same graph completion task. It is observed that **MSGT-GNN** together with other baselines (TINET variants) are generally improving when dimensionality increases from 64 to 256 meanwhile we also notice that the performance become stagnant or starts to drop at large dimension from $d = 256$ to $d = 512$. We hypothesize that low dimensionalities easily falls short of capturing latent features of entities, while high dimensionalities possibly lead to overfitting on the graphs with the increasing

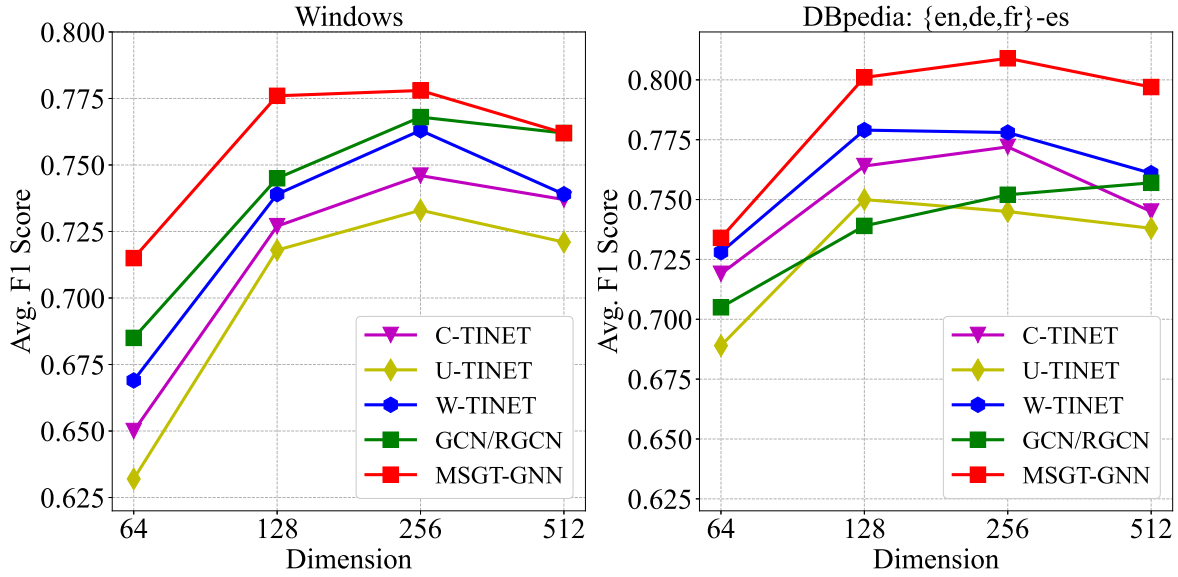


Figure 6.5: Performance comparison with different embedding dimensions d .

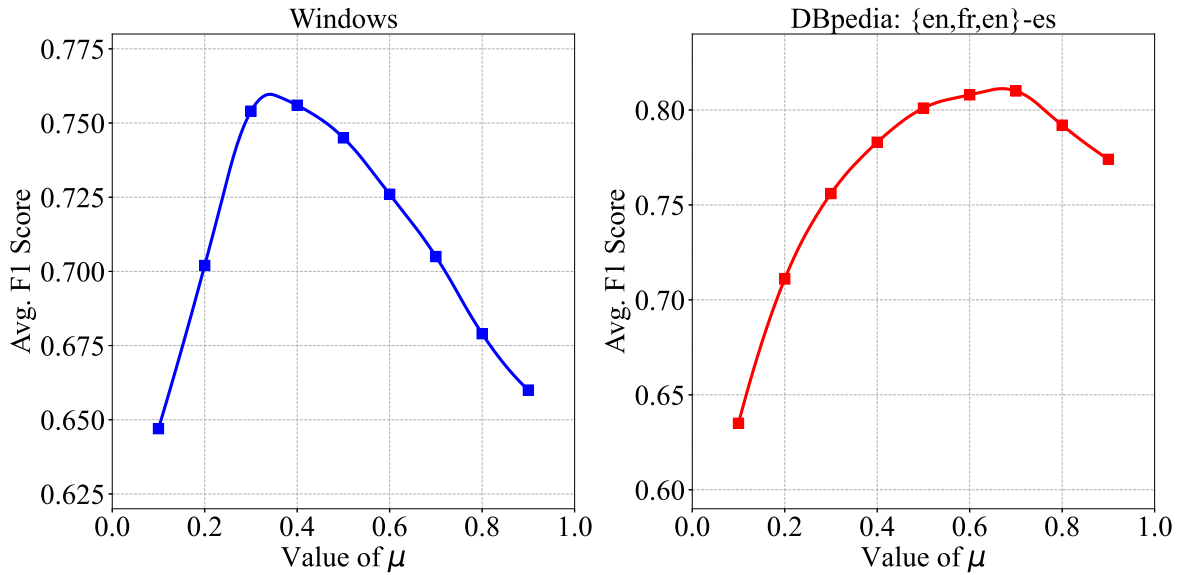


Figure 6.6: Performance comparison with different values of μ , which explicitly balances the importance and reliability of source and target.

model parameters.

Balance between the source and target graphs μ Parameter μ in Equation (8) is another important parameter that significantly affects MSGT-GNN performance. Intuitively, μ controls the leverage between the information from multiple source graphs and from target

graph. In this study, we fix $m = 0.4$ and $d = 128$ for the Windows and Linux dataset and still test on the same graph completion task. As shown in Figure 6.6, choosing μ improperly (either too close to 1 or too close to 0) tends to ignore target domain information or multiple source information, which will compromise the model performance. Also, we notice the optimal μ for different datasets in the experiments are also different. This is largely because the importance of the target information compared to all sources varies in these scenarios. For DBpedia, since the source KBs (en,de,fr) normally contains less conflicting information in multiple sources, the model tends to trust more with $\mu^* = 0.7$ from the given well-developed source graphs.

6.5 Related Work

Transfer Learning, Graph Transfer and Multi-source Adaption Transfer learning, domain adaption, and translation [WKW16] have been widely studied in the past decade and played an important role in real-life applications [SRG16] especially on deep transfer learning [LZW17]. Existing transfer learning research is mostly done on the numeric, grid and sequential data, especially image (specific domain classification, style transfer) and text (translation), but research on graphs, networks, or structured data, whose format are relatively less ordered. Some representative work includes *TrGraph* [FYZ15], which leverages information via common signature subgraphs. [LCT18] is state-of-the-art and most related research aligned with this direction with two-step learning on entity estimation and dependency reconstruction. The aforementioned methods are mostly based on single-graph knowledge transfer. Note that there is some related work on multi-source adaption that has the same goal of reliable knowledge transfer from multiple sources [MMR08]. However, they are still limited within the domain of images and text rather than graphs. Thus their frameworks cannot be directly applied on graph knowledge transfer. Despite the usage of an attention-based model in transfer, one related work [WPZ20] focuses on the node classification task and substantial changes are necessary to make for link prediction in target graphs. We clarify the term of “graph transfer” in Section 6.2 and distinguish it from other research

on the concept of “knowledge transfer” to avoid confusion.

Representation Learning on Knowledge Graphs Graph link prediction is a basic research topic on network analysis. For transfer purposes, [YCZ13] presented a transfer learning algorithm to address the edge sign prediction problem using latent topological features from the target and sources. Collective matrix factorization [SG08] is another major technique. However, these methods are not suitable for dynamics among multiple different domains and the target domain. Another important branch of research related to graph link prediction is network embedding (network representation learning) and similarity search. By representing high-dimensional structured data with embedding vectors, link prediction can be easily performed by node similarity search. These methods can be categorized as meta-path based [SHY11], random walk based [GL16], matrix factorization based [QDM18] and graph neural networks based methods [HYL17a, HLE21]. Similar techniques are applied in multi-relational heterogeneous graphs, i.e. knowledge graphs [VSN19] and their applications [HJC20, HZL20, HLE21]. These embedding based methods (for example [VSN19]) provide insights for representing node features by gathering neighborhood (multi-relational) connections and/or meta-paths and designing graph encoders and decoders. It is worth noted that the most common task over knowledge graphs is triple completion, different from link prediction where focuses on the existence of relations over pairs of nodes in the graph. Another recent research thread along this direction increasingly focuses more on temporal/dynamic graph representation learning [WPC20], which specifically models the graph evolving patterns over time. However, we emphasize that in this work, though it is assumed that the target graphs are relatively incomplete and sparse, we temporarily do not incorporate the time information, as one of the future directions.

Multitask Learning Multitask learning [ZY21] is one emerging active research topic with the rise of artificial intelligence. With the goal of “one model for all tasks”, it is widely applied in the area of computer vision and natural language processing. One of the most common approaches in multitask learning is parameter sharing [Car97]. MSGT-GNNs inspired by the similar multi-task learning mechanism considering each graph as one “task”, however

these frameworks themselves in multitask learning is not applicable for our settings.

6.6 Conclusion and Future work

In this chapter, we formulate a challenging problem on the necessity and benefits of transferring from multi-source graphs into the target graph and then propose **MSGT-GNN**, with the intra-graph Encoder and attention-based cross-graph transfer as major model components. **MSGT-GNN** addresses the challenges and accelerates high-quality knowledge transfer and graph enhancement in the target newly-observed system. Experiments show that **MSGT-GNN** can successfully transfer useful graph knowledge from multiple sources and enable fast target graph construction. For future improvements, one important extension is to temporal graph modeling where we can dive deep into how target graphs grow on newly-deployed systems can grow with the development from multiple sources, which significantly improves explainability on the graph knowledge transfer.

CHAPTER 7

Empowering Homicide Analytics with MurderBook Knowledge Graphs and Domain-specific Language Models

7.1 Introduction

Homicide investigations produce large amounts of text-based data, including basic descriptions of the case, evidence logs, witness interviews, forensic reports and investigator notes. Such information is of vital importance in solving homicides [KJM09, RCO19, PLK21]. In the United States, only around 60% of all homicides reported between 2000-2019 cases were solved or “cleared” through a suspect being taken into custody or through some other circumstance such as the death of the suspect, according to [FBI20]. Homicide clearance rates are thus relatively high compared to other crime types such as burglary, which has a long-standing clearance rate of around 10-15% [Rot17]. Nevertheless, unsolved homicides accumulate year over year, leaving a backlog of cold cases that get harder to solve as time passes. And, given the severe harm of the crime, unsolved homicides create painful burdens on families and communities and drive mistrust in police [JWF16, MFT20]. Conversely, solving homicides contributes to community safety and improves trust in the police [Leo15, BU21].

Analyzing homicide and finding valuable information from large piles of case files remain a difficult task, given the complexity of data from various types and sources. It constantly faces technical challenges to structurize the investigation and lacks model capability and adaptation on crime-related text [PBU20]. However, recent advances in computational

text mining and knowledge discovery [WMW17, QSX20] enable the opportunities to tackle such challenges and offer the potential to improve existing approaches to homicide investigation. We base this supposition on recent research exploring the benefits of constructing and utilizing knowledge graphs in other text-heavy domains including biomedical data mining [HJC20, HLE21], social media modeling [FPW21], and analysis of fictional narratives [AGK20, KET19]. In addition, large language models (LM) [QSX20] and broader foundation models, [BHA21] have revolutionized natural language processing and achieved significant improvement in question answering, document analysis [CFB20], language generation and understanding [GTC21] to the level of human recognition. In one case, [PBU20] used deep learning approaches to construct knowledge graph (KG) representations of evidence, personnel and other information for twenty-four homicide investigations and explored how graph topological features might predict case solvability. Though innovative, this work did not propose a comprehensive framework to bridge the gap between the massive unstructured textual information contained in homicide investigations and ontology-guided knowledge graphs. Such a framework is needed if we are to enable learning for multiple downstream analytical tasks and provide useful insights for homicide investigation teams.

We propose a comprehensive framework, named EIHA, to empower intelligent homicide analysis using crime knowledge graphs and domain-specific language models. The high-level idea is to take advantage of knowledge graphs which can extract important entities and better capture their interactions as facts when crime investigation process develops, and language models which can better portrait and encode text features as widely observed in investigation. Both KG and LM modules mutually enhance our capabilities to analyze crime summaries and lead to critical insights and observations. More specifically, our contributions are 4 folds:

- To our best knowledge, our work is the first to employ the state-of-the-art KG and NLP techniques and investigate their capabilities in learning from homicide investigation to promote community safety and trust in the long run.
- We propose EIHA applied in computational societal crime studies) that are built upon two

major modules: (i) *KG module*, named MKG, builds multi-relational knowledge triples under the guidance of a crime investigation ontology and represents the interactions between persons (e.g., victim, witnesses, etc.) and forms of evidences (e.g., gun, vehicle, etc.); (ii) *LM module*, named M-BERT, is a domain-specific language model, which improves BERT with pretraining and fine-tuning on a large corpus of investigation text;

- We develop a new technique to learn case representations that utilizes hierarchical attentive aggregation from important evidences and case investigation records based on the KG and LM above together towards downstream applications.
- We use EIHA to better solve the case classification task from the learned comprehensive case representations and also demonstrate in case studies that EIHA can be used in a wide range of real-world applications such as semantic search, which creates new opportunities to AI-assisted homicide analysis.

The remainder of this chapter proceeds as follows. In Section 7.2, we outline the homicide investigation and crime knowledge graph (KG) ontologies. In Section 7.3 we describe the creation of MKG M-BERT training, and the combined EIHA for case classification. We provide experimental results in Section 7.4 and application scenarios in Section 7.5. We discuss related works in Section 7.6 and summarize this chapter in Section 7.7.

It is noteworthy that solving homicides improves community safety and contributes to community trust in police [BU21, Leo15, Vau20]. However, solving crime also entails a risk that the wrong individual will be held accountable for a crime they did not commit. Though this risk is relatively low [LHR19, Gar20], it is important to foreground the potential ethical issues entailed in using machine learning methods to assist homicide investigations. We outline limitations and important ethical considerations in the appendix.

7.2 Preliminaries

In this section, we introduce the formulations linking original homicide investigative data to derived knowledge graphs.

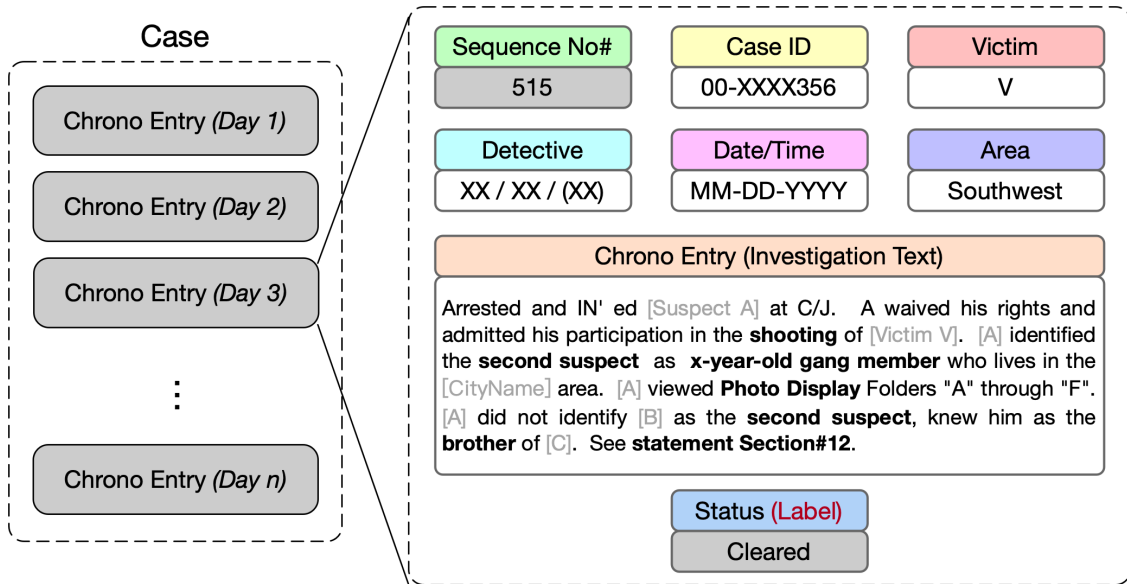


Figure 7.1: A snapshot of one chronological entry from a real-world homicide investigation case.

Data ontology of homicide chronological records In Los Angeles, the vast amount of information generated by a homicide investigation is compiled into a multi-section file colloquially called a “Murder Book” [FJ18]. We are primarily interested in the Murder Book section that records short text descriptions of each sequential step taken by detectives during an investigation. Detectives call this section the “case chronology,” or “chrono” for short. The features in a case chronology include the case identification number, victim and detective names, geographic areas and the police station with jurisdiction, case status such as “cleared” or “open” (considered as case labels), and the content of each investigative step taken by detectives, or a “*chrono entry*” for short. Chrono entries are formatted as tweet-size text statements, which are stored in a tabular format. We target these chrono entries for information extraction and transformation into MKG. Figure 7.1 shows the conceptual structure of a case along with the redacted text of a single chrono entry and its associated features. We discuss data sources, labels and statistics in Section 7.4.1.

MKG: Homicide knowledge graphs The MKG we develop is an example of label-property graph [RWE15]. It is a case-centric, multi-relational representation of a homicide investigation. Named entities (nodes) of multiple types (such as case, entry, evidence, person,

etc) are connected through natural relations (edges) to form factual triplets. Examples of such triplets are {Case#1, hasEntry, Entry#525}, {Case#1, hasDetective, John Doe} and {Entry#525, observeEvidence, Vehicle #Plate}. The ontological or schema of MKG is shown in Figure 7.2, which regulates the structure of our curated knowledge graphs. Note that all information present in the original chrono entry in Figure 7.1 is preserved in the MKG. Some entity types are mentioned in text both by a factual name (e.g., John Doe) and a context-relevant label (e.g., the victim), necessitating both named entity recognition and co-reference determination processes in the KG building stage. Details on MKG extraction, construction and transformation are discussed in Section 7.3.1.

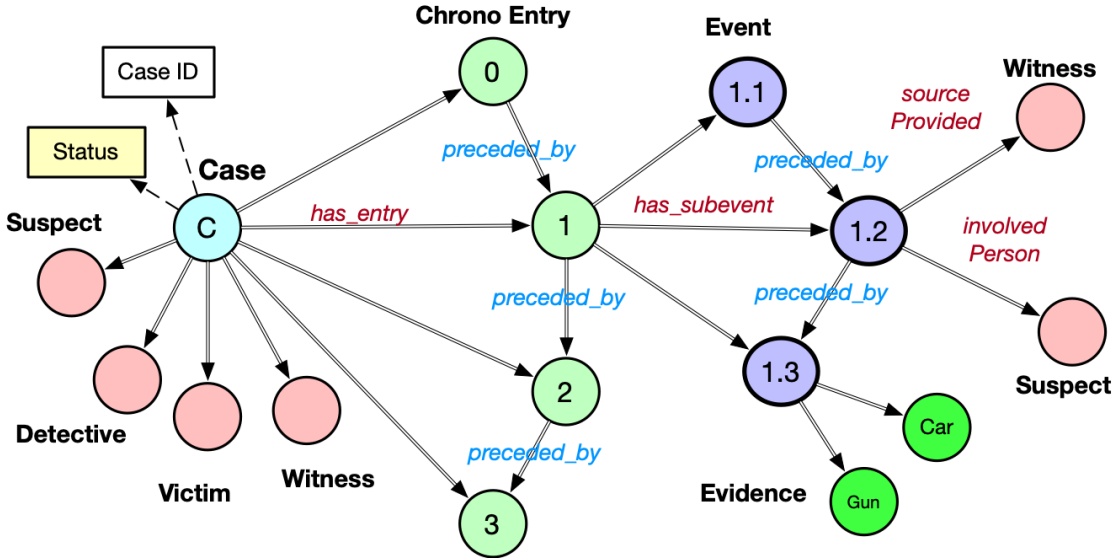


Figure 7.2: Example of the curated case-centric MKG schema with only node types presented.

7.3 Methodology

In this section, we introduce a systematic framework EIHA, designed for intelligent homicide case analytics. The framework has three modules: (1) the MKG module for schema-guided creation and enhancement of knowledge graphs (KG); (2) the M-BERT module to enable better representations of text using a private pretrained language model (LM); and (3) the EIHA, built on top of (1) and (2), to facilitate machine learning applications in homicide

investigations.

7.3.1 MKG: Construction

The main goal of this module is to transform text-based records from the original homicide investigations into structured schema-guided multi-relational knowledge graphs, which is flexible and easy to reuse for continuous studies. It is a data-driven module proposed to tackle with the less structured MurderBook (see Section 7.2), where the chrono entries is less structured and key entities (such as person and evidence) are not extracted and properly modeled. MKG involves entity linkage from external commonsense knowledge bases including ConceptNet [SCH17] and Wikidata [VK14]. The details of MKG implementation pipeline can be partitioned into three stages.

Building ontology/schema MKG is a schema-guided knowledge graph, which enforces a certain ontology in analysis of homicide chronological entries. To serve both knowledge-base search and graph learning tasks, inspired by Reactome [FSG18] data model¹, we establish MKG ontology with entity classes of **Case**, **Entry**, **Event**, **Person**, **Area**, and **Evidence**. Relations are naturally determined by the semantics of entity classes, for example, {**Entity:Event**, **involved**, **Entity:Person**}. Note that we also have necessary subclasses for **Person** including **Suspect**, **Victim**, **Detective** and **Witness**. These reflect important functional roles in homicide cases.

Entity and relation extraction The high-level pipeline of KG construction is inspired by [PBU20], empowered by spaCy [Sri18] with transformer NLP toolboxes such as named entity recognition (NER), co-reference detection and relation extraction ². We highlight the importance of **Evidence** typed nodes which typically represent important non-person objects such as guns and vehicles. In MKG, we build up text-based profiles for every evidence node using text from multiple chronological entries that explicitly mention the related evidence.

¹<https://reactome.org/documentation/data-model>

²Unlike [PBU20], we do not adopt open information extraction (OpenIE). Instead, we strictly follow the defined schema to construct the MKG.

Such entity profiles are used later to help learn case representations. One example is shown in Figure 7.3.

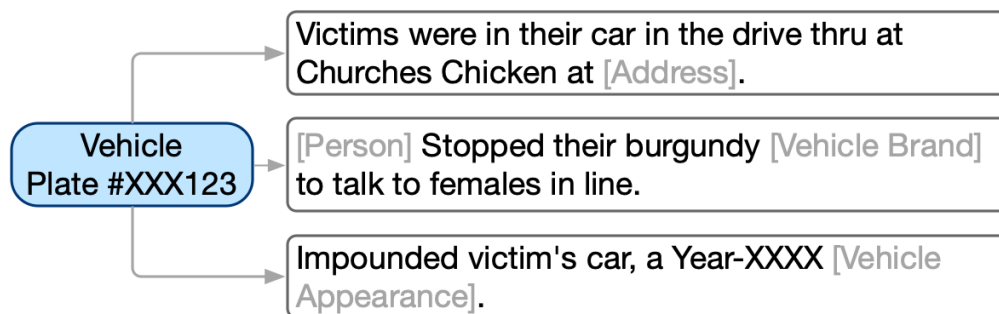


Figure 7.3: Profile example of one (de-identified) vehicle evidence node with extracted descriptions.

Connection to external KGs In our MKG ontology, we also adopt one important entity type `Concept`, which are case-independent nodes sourced from Wikidata [VK14] or ConceptNet [SCH17] entities, such as “gun”, “knife,” or other types of weapon. Our intuition is that these connections may help connect specialized knowledge, internal to the MKG with commonsense external knowledge.

7.3.2 M-BERT: “Crime” Language Model

Inspired by BioBERT [LYK20], a domain-specific language representation model (LM) pretrained on large-scale biomedical corpora, we introduce M-BERT as a large LM for homicide investigation similarly pretrained on unstructured text corpora from our sample of homicide investigations. The corpora include not only the aforementioned chronological entries, but also crime summaries, interviews and more comprehensive investigation details. These data enable a domain-adapted private LM. The motivation of M-BERT as newly developed “crime” language model is that, general-purpose LMs or other existing domain LMs cannot sufficiently model crime-related text created by investigation professionals and police departments.

M-BERT involves both pretraining and fine-tuning stages. We initialize our private LM with the existing, pretrained BERT [VSP17] and undertake further tokenization to deal

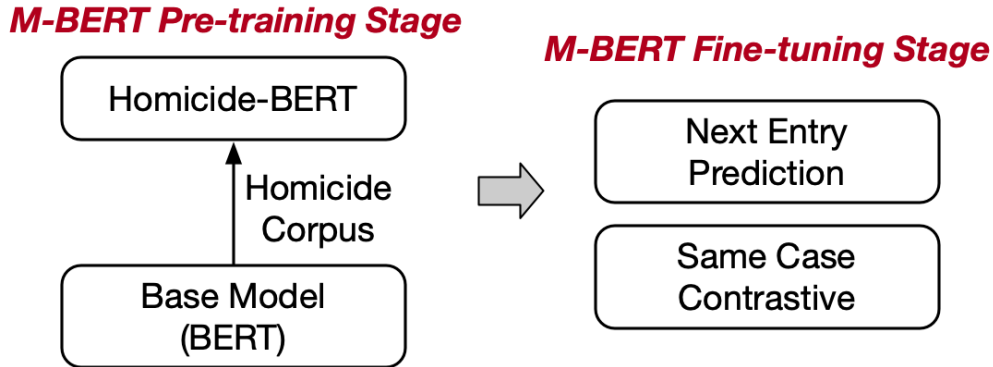


Figure 7.4: M-BERT model workflow for private homicide-domain language models.

with out-of-vocabulary terms, similar to the approach taken with BioBERT [LYK20]. Due to the limited availability of existing named entities relevant for crime corpora (in contrast to that available for biological concepts), we fine-tune M-BERT with *next-entry prediction* and *same-case prediction* tasks based on the chronological nature of investigation process data. The task of next-entry prediction is to classify whether two sequential chrono entries are logically connected via their sequential order, similar to “next sentence prediction.” Same-case prediction seeks to classify whether two selected chrono entries are from the same case. These two tasks are considered a form of self-supervision and therefore enable the M-BERT fine-tuning process. The computational resource we use for training and fine-tuning M-BERT is performed on a 4-core NVIDIA GPU A100-SXM4-40GB (around 100 hours). HuggingFace [WDS19] with the PyTorch [PGM19] library is used throughout M-BERT training and inference.

7.3.3 EIHA: KG-infused Representation Learning Framework

We now discuss how to utilize our two main components, MKG and M-BERT, for downstream applications. We focus on case solvability classification. The goal of case solvability classification is to classify whether any case with current investigative records up to time t is likely to be successfully solved. Our approach to this problem is to first learn case representations as latent features, using MKG and M-BERT in an attention-based encoder. More specifically, the core part of learning case-wise representation is named **Hierarchical**

Attention-based Entry-to-Case Aggregation. It uses M-BERT as the text encoder and selectively aggregates representations of chronological entries with graph entities and relations in the MKG. The module architecture is shown in Figure 7.5.

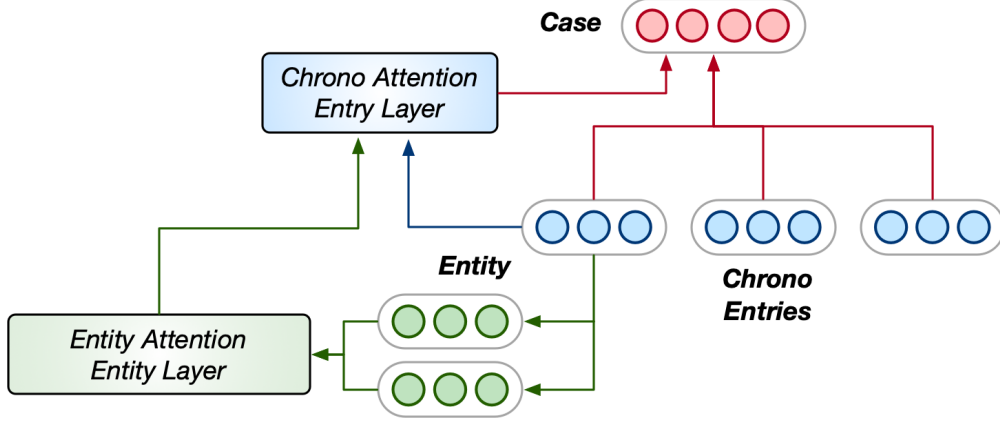


Figure 7.5: Architecture of hierarchical attention layers for case representation learning.

Given a case c with associated observed chronological entries $\{e_i\}$ and attached evidence nodes $\{w_j\}$ ³, our goal is to learn case-wise graph embeddings. We first embed the text attributes of entries and evidence (see Section 7.3.1). We then utilize two parameter-efficient attention layers (evidence- and entry-layers) to hierarchically encode the case features. The first “evidence attention layer” is one of self-attention, where the evidence embedding of one chronological entry i is computed via weighted average, which is,

$$\begin{aligned} \mathbf{E}_w &= \lambda_i \cdot \mathbf{E}_{w_j}, j = 1, \dots, m \\ \lambda_1, \dots, \lambda_m &= \sigma(\lambda_{\text{evi}}^T \mathbf{E}_1, \dots, \lambda_{\text{evi}}^T \mathbf{E}_{w_m}) \end{aligned} \quad (7.1)$$

where $\lambda_{\text{evi}} \in \mathbb{R}^{d_{\text{evi}}}$ (d_{evi} is the dimension of the evidence embeddings). m denotes the total number of evidence nodes observed and $\sigma(\cdot)$ denotes the softmax layer. The second “entry attention layer” takes both the evidence and entry information into consideration as keys. In other words, the attention weights are learned through both the evidence embedding and the entry embedding itself as follows,

³In this stage, we ignore all entries without explicit evidence nodes. Thus, we assume that entries with evidence nodes carry information that is vital to case representation.

$$\begin{aligned} \mathbf{E}_c &= \lambda_i \cdot \mathbf{E}_{e_i}, i = 1, \dots, n \\ \lambda_1, \dots, \lambda_n &= \sigma \left(\lambda_c^T \mathbf{E}'_1, \dots, \lambda_c^T \mathbf{E}'_n \right) \end{aligned} \tag{7.2}$$

where $\lambda_c \in \mathbb{R}^{d_{\text{ent}}+d_{\text{evi}}}$ (d_{ent} and d_{evi} are the dimensions of entry and evidence embeddings). n denotes the total number of chronological entries and $\sigma(\cdot)$ denotes the softmax layer. $\mathbf{E}'_{e_i} = [\mathbf{E}_{e_i} || \mathbf{E}_{w_i}]$ refers to the concatenation of the entry embedding and its attached weighted evidence embeddings. The intuition is that the case representation is selectively learned and aggregated through its entry embeddings as context, while the importance (weight) relies significantly on the associated evidence extracted from MKG. That is, the final case embedding is context-aware of both the chronological entries and important evidence. We point out that the selected structure of attention layers is parameter efficient, considering the limited number of available cases.

Classification After we obtain case representations \mathbf{E} , we use multi-layer neural networks $f_{\text{NN}}(\mathbf{E})$ as the classifier and apply cross-entropy loss on the solved and unsolved cases in the training set, as is standard in most binary classification tasks [MPR05].

7.4 Experiment: Case Classification

7.4.1 Dataset: Source and Statistics

We construct MKG and evaluate our EIHA based on samples of Murder Books housed in the Los Angeles Police Department’s Homicide Library [FJ18]⁴. The sample includes 490 selected homicide cases from LAPD’s South Bureau spanning the years 1990-2010. The 490 cases produced 27,518 discrete chronological entries with an average of 56.2 entries per case and 44.1 words per entry (after removing punctuation and stop words). The first entry of a case chronology is always the notification of the detectives of the location of a homicide. The last entry reflects the most recent action taken on the case, which is dependent upon

⁴These data are not publicly available due to their extreme sensitivity. Research was conducted under UCLA IRB Protocol #19-000588

its status. For example, for an open case, the last entry is the result of the most recent six-month open case review. All cases are labeled as {0: Open, 1: Cleared, 2: Cleared (other)}⁵. Cases with the labels of {1,2} are considered as “Solved” (237 cases) while {0} as “Unsolved” (224 cases) to facilitate binary classification.

It is worth emphasizing that the number of chronological entries in a Murder Book naturally grows in time with the investigation processes. We anticipate, therefore, that the available information (descriptions, interviews and evidence) in MKG should be different as time t increases from the date of the incident. Figure 7.6 shows the time trend in the number of chronological entries. The pattern suggests that we need to control for the time in the classification task (see Section 7.4.2 for details).

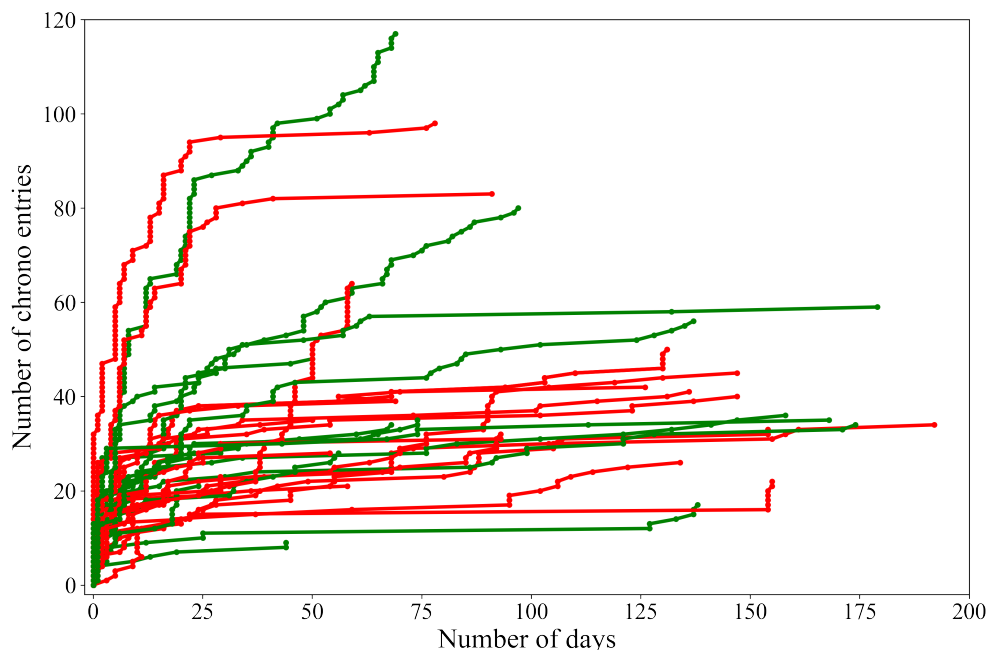


Figure 7.6: Number of chronological entries vs case duration in days since the incident for 40 randomly selected cases. Green/red lines denotes solved/unsolved cases respectively. (Best viewed in color)

The unstructured text used in Section 7.3.2 comes from all homicide investigation materials including both chrono entries and other supplementary materials such as interviews

⁵“Cleared (other)” indicates that cases were closed by some means other than arrest such as “suspect deceased”.

and crime summaries, which combined add up to more than 1.24G tokens.

7.4.2 Experimental Setup

Evaluation protocol We formulate our experiment as a binary classification task. Following [PBU20], we report F1 score, AUROC and false discovery rate (FDR) as evaluation metrics. Because the input of chrono entries in one case is time-sensitive, we restrict the input information to a fixed time window (i.e., 14 days from the incident occurrence time). This configuration applies to both the input of textual chronological entries and MKG entities, for all models. Best performed models are obtained through cross validation and tested on the test set (20% of total cases that are randomly selected). All classifiers are implemented with `scikit-learn` [BLB13].

Baseline Methods We consider the following baseline approaches (with hyperparameters) from our own EIHA. The baselines can be categorized as conventional classifiers on graph metrics, based only on the MKG, and classifiers based on document embeddings.

- **LR-Basic** Simple logistic regression used by [PBU20]. The model is formulated as $p(y = 1|\mathbf{x}) / (1 + \exp(-\beta\mathbf{x}))$ where $\beta\mathbf{x} = \beta_0 + \beta_1 \cdot s + \beta_2 \cdot e + \beta_3 \cdot (s \cdot e)$, where s and e are number of suspect nodes and evidence nodes. Note that the features used are restricted to a given time window;
- **LR-Extend** Similar to LR-Basic, we apply logistic regression with extended features on more node types (such as chrono entry, person, etc.);
- **SVM/RF/NN-Classifier** We use other representative classifiers such as SVM (default RBF kernel, $C = 1.0$), random forest and 2-layer neural network with hidden dimensions $\{d_1, d_2\} = \{128, 32\}$ on the same features as in LR-Extend;
- **Doc2vec** We obtain fixed-length feature representations of cases as documents for classification by Doc2vec [LM14], implemented by `genism`.
- **SentBERT, XLM-RoBERTa, GPT-2** We concatenate the text of all input chrono entries as a document with encoders such as [RG19], [CKG20] and [RWC19] (no training)

with 2-layer NN as classifier (hidden dimensions set as $\{d_1, d_2\} = \{128, 32\}$);

- **M-BERT** We replace SentBERT, XLM-RoBERTa, GPT-2 with our private language model described in Section 7.3.2. The same 2-layer NN classifier is applied.

Variant Models We also explore the following variants of EIHA. Note that dimensions of all variants are set as $d_{\text{evi}} = 128$ and $d_{\text{ent}} = 368$ and 2-layer NN as classifier (hidden dimensions as $\{d_1, d_2\} = \{128, 32\}$).

- **EIHA**: Proposed model in Section 7.3.3;
- **EIHA(avg)**: Excludes all attention layers and applies average operations on entity and entry embeddings;
- **EIHA(no-evd)**: Excludes all evidence nodes with its associated descriptions and only applies the attention layer on entries;
- **EIHA(select)**: Manually selected top-5 entries with the highest number of evidence nodes, instead of all entries with evidence nodes.

7.4.3 Results

The main results of case classification are shown in Table 7.1. We observe that EIHA improves case classification by 4.0% (a relative gain of 5.4% on F-1 score (6.1% on FDR), compared with best performing graph-embedding classifier (RF), and by 3.6% over the best performing document embedding models (M-BERT). The results suggest that it is effective in practice to combine language models and knowledge graphs in the classification task, compared to approaches that use LM or KG only. Our customized homicide-related LM M-BERT outperforms other LMs obtained from a general corpus on all classification metrics. We hypothesize that domain-adapted LM from homicide records has a better capability of modeling crime-related chronological entries.

As one ablation study, we also compare the case classification results with several model variants. Compared to the standard configuration, both **avg** and **no-evi** variants perform

significantly worse, while the **select** model variant achieves comparable F1 scores in the classification task. These results suggest that an attention mechanism that selectively learns from the top “influential” entries and items of evidence helps produce more comprehensive and representative embeddings. We also find that the combination of attention layers performs better among all model variants.

Table 7.1: Results of case classification, comparing two categories of approaches. All shaded methods are developed in this work and the best results are **bolded**.

Methods	AUROC	F1	FDR
LR-Basic	0.917 ± 0.006	0.607 ± 0.012	0.612 ± 0.012
LR-Extend	0.936 ± 0.008	0.686 ± 0.009	0.653 ± 0.012
SVM-Classifier	0.945 ± 0.005	0.703 ± 0.014	0.653 ± 0.018
RF-Classifier	0.953 ± 0.005	0.742 ± 0.015	0.714 ± 0.015
NN-Classifier	0.945 ± 0.011	0.704 ± 0.017	0.632 ± 0.016
Doc2vec	0.922 ± 0.008	0.645 ± 0.005	0.591 ± 0.018
SentBERT	0.949 ± 0.010	0.726 ± 0.012	0.673 ± 0.020
XLM-RoBERTa	0.944 ± 0.009	0.701 ± 0.013	0.714 ± 0.016
GPT-2	0.952 ± 0.009	0.736 ± 0.019	0.734 ± 0.010
M-BERT	0.957 ± 0.011	0.746 ± 0.022	0.755 ± 0.022
EIHA	0.962 ± 0.018	0.782 ± 0.020	0.795 ± 0.024
EIHA(avg)	0.954 ± 0.015	0.740 ± 0.019	0.755 ± 0.020
EIHA(no-evd)	0.957 ± 0.009	0.751 ± 0.022	0.755 ± 0.025
EIHA(select)	0.963 ± 0.019	0.778 ± 0.026	0.795 ± 0.024

7.5 Applications

EIHA is proposed as a generic framework that may enable multiple downstream applications beyond case classification. In this section, we demonstrate applications in two additional tasks.

Semantic search on chronological entries and homicide cases As shown in the case classification task, our proposed EIHA can produce both entry-wise and case-wise embeddings. In practice, such embeddings contain semantic information including possibly common homicide “themes,” weapon usage patterns, and incident locations, that can be further used to support “semantic search.” Table 7.2 shows two examples of semantic search over chrono-

logical entries. Specifically, in Query 1, a search for the top three most similar entries to a target entry all have common themes of “hospitalization” and “victim announced death”, together with an explicit connection to “Southwest Station.” Similarities thus encompass both conceptual semantics and specific crime details. Similar broad semantic connections can be seen in the returns from Query #2.

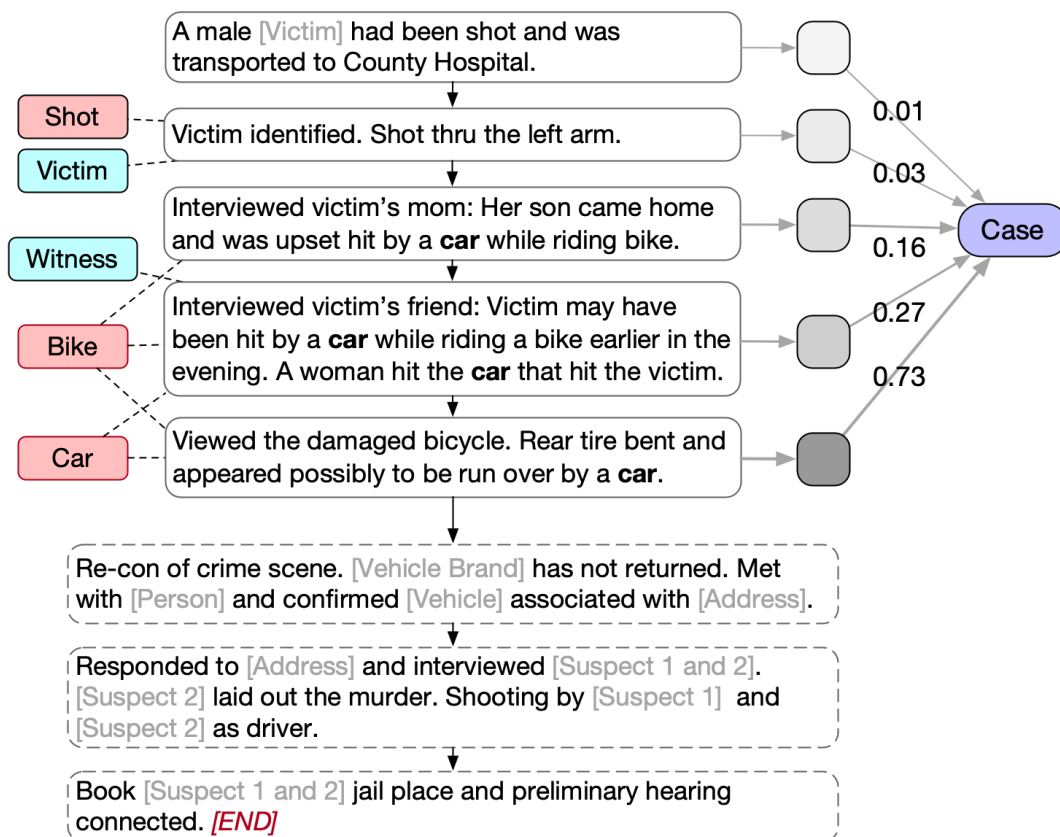


Figure 7.7: Visualization of attention weights from multiple input entries on the classification of one single case. Given the first 5 entries as input, the entry with the observation of *Car* at the crime scene leads to the highest weight predicted by EIHA, instead of *Bike*. The last 3 chrono entries are provided for the full context of the case, but not as model input.

Attention Visualization and Important Chrono Entry Highlighting As shown in Section 7.3.3, the attention learned by EIHA can reflect the importance of each input chrono entry and evidence. In other words, we can highlight the chronological entries that are associated with higher weights in case embeddings as shown in Equation 7.2, which may yield important insights into the investigative process. We show an example of one case with relatively few chrono entries in Figure 7.7. In this hit-and-run homicide, multiple types

of evidence appear as inputs in the first five chronological entries such as the bike, missing rental car, and the deadly gunshot. EIHA learns that the top-ranked chrono entries mostly include the car evidence, suggesting that this evidence was critical in solving the case in the end. A similar strategy can also be applied to the entry-evidence self-attention layer to highlight potential high-value evidences that might accelerate the investigation.

7.6 Related Work

Our efforts must be understood in the context of related work on knowledge graphs, language models in NLP, and sociological studies of crime and homicide.

Pillar 1: KG Construction and representation learning. Knowledge graphs (KGs) are a type of multi-relational graph data structure. Well-known examples include DBpedia [LIJ15], YAGO [PWS20], and many other domain-specific KGs [HJC20]. Knowledge graph creation and curation [WDR21], and KG representation learning are foundations in a wide range of knowledge-driven downstream AI applications including translational- or similarity-based embedding [WMW17, HCY19] and graph neural networks [VCC18], which are vital to capturing the latent semantic features and support relational inferences (i.e. link prediction). Graph attention is also central to building graph representations that selectively learn entities and relations [WHC19, HLE21].

Pillar 2: Advances on language models and applications. Transformer-based, large pre-trained language models (LMs) [QX20], originated from BERT [VSP17], have been widely applied in NLP as a revolutionary milestone, such as text generation, natural language inference and document classification [CFB20, ZSW22]. Current research makes use of pre-trained language models for applications involving low-resource languages, such as AraBERT [ABH20], and domain-specific NLP applications, like in biomedical research, with pioneering work as BioBERT [LYK20] and BlueBERT [PYL19]. Our proposed EIHA aligns as domain-specific LMs in criminology for document-level analysis.

Pillar 3: Computational social science. Crime has been a prominent topic within the

social sciences for more than a century. Criminology as a discipline has naturally divided itself into domains studying offenders (and sometimes victims), or events [Wor10]. The process of police investigations has generally been studied from the point of view of events and the contexts in which they occur. Computational criminology is a relatively young subfield that has focused mostly on modeling and measuring geographic crime patterns [Bra11] or offender recidivism [Ber13]. Some recent work has started to incorporate natural language processing techniques into crime analysis [KBB17]. The use of knowledge graphs in crime analysis is relatively new. Related efforts include [PBU20, QW17, ST19, AGK20]. Our work aligns with this direction that bridges NLP and KG to improve crime analytics and prevention.

7.7 Conclusion and Future Directions

In this chapter, we introduce an innovative framework named EIHA, the first to apply the state-of-art technology in Graph ML and NLP to homicide analytics and crime prevention. It is collectively based upon both domain knowledge graph MKG and domain language model M-BERT. MKG, as a KG backend, is constructed from homicide investigation record and provide structured and multi-relational data formulation. In addition, M-BERT, a private domain language model obtained through pre-training and fine-tuning on the homicide corpus, can better encode textual features and be integrated into deep learning systems. In this work, we investigate EIHA on case solvability classification task, and explore two application scenarios with the power of EIHA, which bridges the gap between graph and NLP in crime studies and has shown effective and outperform multiple state-of-the-art methods. We also point out AI-assisted analytics on criminology urge interdisciplinary academic and industrial communities to collectively tackle challenges in an attempt to build socially responsible and inclusive AI systems and applications.

7.8 Limitations

One limitation in our proposed EIHA is brought by the customization on data and modeling on one specific domain of crime and homicide reports. For example, MKG is a data-driven schema established based on the original MurderBook record, which may not be easily extended to applications from other domains. Consequently, another layer of limitation is from the data availability and accessibility to labels. While EIHA favors more data to fine-tune language models and KG construction, and more labels for supervision in training on downstream tasks, it requires human expertise to obtain them through real-life case report. This inspires one of our future directions as parameter-efficient models and self-supervision strategies. In addition, EIHA has pretrained large language models as one foundational component, along with its limitations [TBC21] on scaling, training resources, potential bias and privacy concerns. In terms of potential limitations regarding broader social impact, we discuss under the following section of “Ethics Statement and Broader Impact”. Last but not least, case representation as dense embedding vectors in EIHA applications can be examined and improved on explainability and further explored on its causal indications.

7.9 Ethical Considerations and Broader Impact

This research was conducted under UCLA IRB protocol #19-000588. . The data are not publicly available due to their high sensitivity and inclusion of private and confidential information. These constraints create challenges for external validation of our approach. Nevertheless, transparent implementation and testing guidelines should make it possible for others to build and study similar knowledge graphs for homicide investigations in other private settings.

Our two proposed downstream uses of EIHA include basic search functionality and classification tasks. Both are impacted by the quantity and quality of data contained in the corpus of investigative texts as well as the choices made to facilitate the construction of a knowledge graph from raw text. We are working with a small sample of homicides relative

to the total for Los Angeles over the study period from 1990-2010. Thus, the KGs and LMs presented here should not be considered representative of homicide investigations overall.

Homicide investigations certainly incorporate biases. There is little evidence at present that “victim devaluing” based on extralegal factors such as race play any substantial role in determining the outcomes of homicide investigations [RJM20, Vau20]. Nevertheless, we must recognize that homicide knowledge graphs likely contain (and lack) some entities/relations as a result of investigative bias. Rather than being a basis for rejecting this approach, we suggest that a careful comparative study of how knowledge graphs differ across detectives and crime contexts might help identify how such biases operate and provide a pathway for their correction.

The potential future use of EIHA to augment homicide investigations presents a unique set of ethical challenges. Such augmentation may improve the efficacy and fairness of homicide investigation beyond what is capable with current procedures and technologies. However, it is the potential that machine learning methods introduce or compound existing biases that is of greatest concern. For example, if the experiments on case classification presented here were to move to field deployment, one potential risk is that a machine-classification of a case as unlikely to be solved, early in the investigative process, might lead detectives to shift their efforts elsewhere. Official policy and procedure that requires detectives to redouble their efforts in response to such a classification would be one corrective that could also improve case solvability, beyond what would have otherwise been the case.

Other potential extensions of knowledge graphs for homicide investigations raise additional concerns. Homicide knowledge graphs might also be compared to identify what makes some cases solvable and others not. Some differences between cases might simply highlight what is likely already obvious to investigators. For example, the absence of DNA evidence in one case may be an obvious “cause” for that case going unsolved. The absence of a DNA evidence node in the corresponding graph does not create any “surprises.” By contrast, if there are non-intuitive “causal pathways” in graphs for solved cases that are absent in unsolved cases, then the potential surprise of this finding could alter the way that homicide investi-

gations unfold. How criminal and procedural law would handle novel investigative practices based on analytical “surprises” is an open question [BZM21]. Method transparency is central to understanding such surprises and the effects [RWC20].

Recent research has highlighted several ethical and social risks associated with the use of LMs, especially trained from texts that involves personal identifying information [WMR21]. In this work, we mitigate potential information leaks by masking all sensitive person-related attributes in M-BERT training. The comparative approach we suggest for identifying bias in knowledge graphs might also be used here to identify potential biases emanating from the language models.

It is widely agreed that a failure to solve homicides, particularly in low-income, minority communities, contributes to concentrated disadvantage and erodes community trust. Yet, adopting an “ends-justify-the-means” approach to solving homicides may generate the opposite effects. In conclusion, machine learning methods must not only add scientific and practical value to the homicide investigative process, but must do so while adhering to important legal and ethical principles [Chi19].

Table 7.2: Top-3 similar results retrieved by EIHA given one chrono entry in MurderBook. Tokens that are potentially highly related are shaded. Sensitive information have been anonymized or redacted.

Query Chrono entry: Hospital death, Southwest	
(Detective-1) Southwest Station, Notified by Detective III ■■■■ of a hospital death. ■■■■ expired at USCMC.	
Rank	Similar Chrono entries
#1	(Detective-1&2) Southwest Station Briefed at Southwest Station. Obtained printouts on Incidents 0022 and 0521. Victim was pronounced dead at Kaiser WLA .
#2	(Detective-1,2&3) Southwest Station Arrived at Southwest Station. Briefed by ■■■■ on case. Victim was pronounced dead at 1750 hours.
#3	(Detective-1&2) Southwest Station Received update on shooting. ■■■■ died at 1615 hours. Obtained incident history printout. Possible suspect was detained after ■■■■ saw him running away from crime scene minutes after the shooting.

(a) Query #1 on the plot of Hospital death, Southwest and its retrieved similar entries

Query Chrono entry: Autopsy, Gunshot, Wound	
Detective-4 received phone call from Doctor-1's from L.A. Officer-1's office. Doctor-1 conducted autopsy on Suspect. He ascribed the cause of death as a single gunshot wound to the head. Stippling was present at entry wound.	
Rank	Similar Chrono entries
#1	■■■■ contacted Doctor-2, Pathologist L.A. County Officer-1's office regarding the results of the post-mortem examination . Doctor-2 advised that the suspect sustained a single through-and-through tight contact gunshot wound to the right side of his head .
#2	■■■■ attended the autopsy performed by Doctor-3. He ascribed the cause of death as a single gunshot to ■■■■'s lower left back . He recovered 2 bullets and the jacketing to one of them.
#3	Detective-5 attended autopsy with Doctor-4. Cause of death ruled as single gunshot wound through the back of victim's head . (1) bullet recovered from victim's left upper leg (medium caliber).

(b) Query #2 on the plot of Autopsy, Gunshot, Wound its retrieved similar entries

CHAPTER 8

Conclusion

In this dissertation, we have introduced multiple works that focus on how ontological information is presented in different knowledge graphs as internal hierarchical structures and how to incorporate such information in knowledge graph representation learning via various strategies such as two-view joint learning (Chapter 2 and 4), weighted association with taxonomies (Chapter 3), hybrid modeling of relational and hyperbolic graph neural networks 5, cross-graph attention fusion in graph encoder 6. The benefits of incorporating ontological information into knowledge graph modeling has been demonstrated by multiple tasks and applications, with representative examples as in Chapter 3 and 5 (Bioinformatics and healthcare), 4 (E-Commerce) and 7 (Social and crime prevention).

The contributions of this dissertation in advancing deep learning in knowledge graph (Graph ML) and KG-empowered applications can be further summarized as follows:

- Ontological information in different knowledge graphs has various ways of formulations, which are possibly unlike with each other. While some ontologies are formally well-defined (such as product departments, semantic web, disease, and gene ontology) to portray the relationship between classes and internal structures, there are many situations where additional efforts are required for a thorough inspection of the knowledge graph we have and the applications we aim at. We have shown in previous chapters how the knowledge graphs are formulated by observing how entities and relations are semantically defined and statistically observed. Such process in the lifecycle of data mining and machine learning are generally valuable since it helps us understand the data and provides more analytical insights.

- Incorporating ontological information into learning knowledge graph representations is significantly beneficial. We have illustrated that our dedicated strategies such as multi-task learning framework with the ontological view of the KGs, adding reasonable auxiliary learning objectives on ontologies, attention mechanism on entities and relational facts inside the KGs have resulted in better performance on various tasks. Along with many state-of-the-art methods transferred including, but not limited to, the large family of graph neural networks, Transformer, and language models, these approaches provide a collection of powerful toolboxes and enable advantageous modeling over hierarchical knowledge graph data and learn more comprehensive representations.
- Effectively leveraging ontological information and KG structures can help tackle many challenging research problems. Since many ontologies provides representative information on how entities and relations are organized and categorized, they are extremely useful to solve cold-start problems, low-resource items and inductive scenarios, as demonstrated in ontology population (Chapter 2), protein interaction and target prediction in emerging disease (Chapter 3), recommendations for cold-start shopping products (Chapter 4), newly deployed system activity monitoring (Chapter 6) and domain-specific knowledge graph construction (Chapter 7).
- We have noticed strong capabilities and encouraging results by using knowledge graphs to empower interdisciplinary applications. Our research work in this dissertation has covered a relatively large range of applications domains in biological and biomedical knowledge graphs (Chapter 3 and 5), product graphs and complementary recommendation (4), homicide investigation and crime prevention. We can further expand our vision of knowledge graphs to promote more knowledge-intensive applications in public health, personalized healthcare, e-commerce, and intelligent systems for social good.

In the end, this is by no means the end of the exploration of the power of knowledge graphs in machine learning and general artificial intelligence. We also propose a few directions for future work:

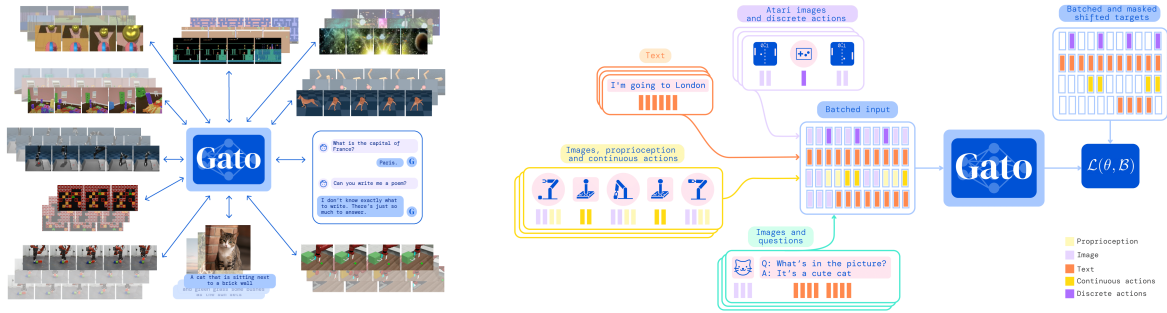


Figure 8.1: Inspired by progress in large-scale language modeling, DeepMind has proposed a multi-modal, multi-task, multi-embodiment generalist AI agent that can serve multiple inputs of images, text, and human-robot interactions and corresponding applications, ranging from gaming, chatbots, visual question answering.

- Universal framework of graph machine learning.** As we have shown in previous chapters, knowledge graphs in real life are based on various formulations in different domains, which results in a large difference in the semantic structures. Inspired by Foundation Models [BHA21], one universal graph learning mechanism that can adapt to the variances of KG structures, potentially ranging from encyclopedia-like KGs, biological ontologies and healthcare records, social media interactions, and many more, can reduce the modeling effort and results in new emergent capabilities.
- Mutual augmentation of knowledge graphs and natural language processing** As knowledge graph naturally connects to natural language processing in multiple applications, it is believed that KG as structured knowledge can be extracted and distilled from the unstructured text as a factual reflection from the real world [SWF21]; and in turn, many downstream tasks such as summarization, entity recognition, question answering and language generation can positively be affected by high-quality KG [ZXR22]. KG and NLP are mutually complementary to each other and learning augmentation techniques can significantly benefit a wide range of real-world applications.
- KG-incorporated Multi-modality Learning, together with vision and language for more powerful and widespread Artificial General Intelligence.**

Considering one example named GATO [RZP22] shown in which was proposed by DeepMind, Figure 8.1, Artificial General Intelligence (AGI) has entered the spotlight in the AI/ML communities as one of the next promising frontiers. Knowledge graphs as graph-mode data can uniquely contribute to existing research on multi-modality research, combining and integrating the modality of vision, language and audio. Researchers have already expanded similar opportunities in this trend such as incorporating commonsense knowledge (such as scene graphs) to enhance vision-language representations [YTY21]. Knowledge graph will further expand its new capabilities in promoting research in AGI and make AI applications more logical and understandable, transparent and explainable, and socially responsible and beneficial.

BIBLIOGRAPHY

- [ABH20] Wissam Antoun, Fady Baly, and Hazem Hajj. “AraBERT: Transformer-based Model for Arabic Language Understanding.” In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 9–15, 2020. 163
- [ABK07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. “Dbpedia: A nucleus for a web of open data.” In *The semantic web*, pp. 722–735. Springer, 2007. 128, 137
- [AGK20] Mariam Alaverdian, William Gilroy, Veronica Kirgios, Xia Li, Carolina Matuk, Daniel McKenzie, Tachin Ruangriengsin, Andrea L Bertozzi, and P Jeffrey Brantingham. “Who killed Lilly Kane? A case study in applying knowledge graphs to crime fiction.” In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 2508–2512. IEEE, 2020. 4, 149, 164
- [AH11] Dean Allemang and James Hendler. *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Elsevier, 2011. 1
- [AK14] Tevfik Aytekin and Mahmut Özge Karakaya. “Clustering-based diversity improvement in top-N recommendation.” *Journal of Intelligent Information Systems*, **42**(1):1–18, 2014. 101, 102
- [AKM17] Mona Alshahrani, Mohammad Asif Khan, Omar Maddouri, Akira R Kinjo, Núria Queralt-Rosinach, and Robert Hoehndorf. “Neuro-symbolic representation learning on biological knowledge graphs.” *Bioinformatics*, **33**(17):2723–2730, 2017. 44
- [AOO20] Tareq Al-Moslmi, Marc Gallofré Ocaña, Andreas L Opdahl, and Csaba Veres. “Named entity extraction for knowledge graphs: A literature overview.” *IEEE Access*, **8**:32862–32881, 2020. 4

- [BB14] Volha Bryl and Christian Bizer. “Learning conflict resolution strategies for cross-language wikipedia data fusion.” In *Proceedings of the 23rd International Conference on World Wide Web*, pp. 1129–1134, 2014. 42
- [BBS17] Arijit Biswas, Mukul Bhutani, and Subhajit Sanyal. “Mrnet-product2vec: A multi-task recurrent neural network for product embeddings.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 153–165. Springer, 2017. 73
- [BEP08] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. “Freebase: a collaboratively created graph database for structuring human knowledge.” In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250. AcM, 2008. 1
- [Ber13] Richard Berk. “Algorithmic criminology.” *Security Informatics*, **2**(1):5, 2013. 164
- [BGW14] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. “A semantic matching energy function for learning with multi-relational data.” *Machine Learning*, **94**(2):233–259, 2014. 23
- [BHA21] Rishi Bommasani, Drew Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney Arx, Michael Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, and Percy Liang. “On the Opportunities and Risks of Foundation Models.” *arXiv preprint arXiv:2108.07258*, 2021. 149, 171
- [BHN07] Helen Berman, Kim Henrick, Haruki Nakamura, and John L Markley. “The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data.” *Nucleic acids research*, **35**(suppl.1):D301–D303, 2007. 40

- [BK16] Oren Barkan and Noam Koenigstein. “Item2vec: neural item embedding for collaborative filtering.” In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2016. [73](#), [101](#)
- [BLB13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. “API design for machine learning software: experiences from the scikit-learn project.” In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013. [159](#)
- [BN09] Jens Bleiholder and Felix Naumann. “Data fusion.” *ACM computing surveys (CSUR)*, **41**(1):1–41, 2009. [42](#)
- [Bra11] Patricia L Brantingham. “Computational criminology.” In *2011 European Intelligence and Security Informatics Conference*, pp. 3–3. IEEE, 2011. [164](#)
- [BU21] P. Jeffrey Brantingham and Craig D. Uchida. “Public cooperation and the police: Do calls-for-service increase after homicides?” *Journal of Criminal Justice*, **73**:101785, 2021. [148](#), [150](#)
- [BUG13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. “Translating embeddings for modeling multi-relational data.” In *Advances in neural information processing systems*, pp. 2787–2795, 2013. [3](#), [13](#), [19](#), [22](#), [23](#), [24](#), [25](#), [36](#), [47](#), [52](#), [102](#)
- [BZM21] Jane R Bambauer, Tal Zarsky, and Jonathan Mayer. “When a Small Change Makes a Big Difference: Algorithmic Fairness Among Similar Individuals.” *UC Davis Law Review, Forthcoming, Arizona Legal Studies Discussion Paper*, 2021. [167](#)
- [BZS19] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and

- Jun Gao. “Personalized Bundle List Recommendation.” In *The World Wide Web Conference*, pp. 60–71. ACM, 2019. [101](#)
- [CAP20] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. “Machine Learning on Graphs: A Model and Comprehensive Taxonomy.” *CoRR*, [abs/2005.03675](#), 2020. [108](#)
- [Car97] Rich Caruana. “Multitask learning.” *Machine learning*, **28**(1):41–75, 1997. [146](#)
- [CFB20] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. “SPECTER: Document-level Representation Learning using Citation-informed Transformers.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2270–2282, 2020. [149](#), [163](#)
- [CGH18] Chen Chen, Behzad Golshan, Alon Y. Halevy, Wang-Chiew Tan, and AnHai Doan. “BigGorilla: An Open-Source Ecosystem for Data Preparation and Integration.” *IEEE Data Eng. Bull.*, **41**(2):10–22, 2018. [127](#)
- [Chi19] Vincent Chiao. “Fairness, accountability and transparency: notes on algorithmic decision-making in criminal justice.” *International Journal of Law in Context*, **15**(2):126–139, 2019. [167](#)
- [CHS04] Sherri de Coronado, Margaret W. Haber, Nicholas Sioutos, Mark S. Tuttle, and Lawrence W. Wright. “NCI Thesaurus: Using Science-Based Terminology to Integrate Cancer Research Results.” In *MEDINFO*, volume 107, pp. 33–37, 2004. [119](#)
- [CJZ19] Muhao Chen, Chelsea J-T Ju, Guangyu Zhou, Xuelu Chen, Tianran Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. “Multifaceted protein–protein interaction prediction based on Siamese residual RCNN.” *Bioinformatics*, **35**(14):i305–i314, 2019. [44](#), [45](#)
- [CKG20] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer,

- and Veselin Stoyanov. “Unsupervised Cross-lingual Representation Learning at Scale.” In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, 2020. [159](#)
- [Con18] Gene Ontology Consortium. “The Gene Ontology Resource: 20 years and still GOing strong.” *Nucleic acids research*, **47**(D1):D330–D338, 2018. [40](#), [54](#)
- [CS04] Aron Culotta and Jeffrey Sorensen. “Dependency tree kernels for relation extraction.” In *Proceedings of the 42nd annual meeting on association for computational linguistics*, p. 423. Association for Computational Linguistics, 2004. [14](#)
- [CTC18a] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. “Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment.” In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3998–4004. AAAI Press, 2018. [14](#), [44](#)
- [CTC18b] Muhao Chen, Yingtao Tian, Xuelu Chen, Zijun Xue, and Carlo Zaniolo. “On2vec: Embedding-based relation prediction for ontology population.” In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 315–323. SIAM, 2018. [14](#), [49](#)
- [CTY17] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. “Multilingual knowledge graph embeddings for cross-lingual knowledge alignment.” In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1511–1517. AAAI Press, 2017. [14](#), [28](#), [44](#), [120](#)
- [CYR19] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. “Hyperbolic graph convolutional neural networks.” *Advances in neural information processing systems*, **32**, 2019. [4](#), [108](#), [114](#), [115](#), [126](#)
- [CZC16] Wei Cheng, Kai Zhang, Haifeng Chen, Guofei Jiang, Zhengzhang Chen, and Wei Wang. “Ranking causal anomalies via temporal and dynamical analysis on

- vanishing correlations.” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 805–814, 2016. 128
- [CZL17] Lijun Chang, Chen Zhang, Xuemin Lin, and Lu Qin. “Scalable Top-K structural diversity search.” In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 95–98. IEEE, 2017. 102
- [DCL19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. 113
- [DCW17] Boxiang Dong, Zhengzhang Chen, Hui Wang, Lu-An Tang, Kai Zhang, Ying Lin, Zhichun Li, and Haifeng Chen. “Efficient discovery of abnormal event sequences in enterprise security systems.” In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 707–715, 2017. 128
- [DHI12] AnHai Doan, Alon Y. Halevy, and Zachary G. Ives. *Principles of Data Integration*. Morgan Kaufmann, 2012. 127
- [DHW21] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. “Heterogeneous network representation learning.” In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 4861–4867, 2021. 126
- [DMS18] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. “Convolutional 2d knowledge graph embeddings.” In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 13

- [Don06] Kevin Donnelly. “SNOMED-CT: The advanced terminology and coding system for eHealth.” In *Stud Health Technol Inform*, volume 121, pp. 279–290, 2006. 119
- [DQW17] Jianfeng Du, Kunxun Qi, Hai Wan, Bo Peng, Shengbin Lu, and Yuming Shen. “Enhancing knowledge graph embedding from a logical perspective.” In *Joint International Semantic Technology Conference*, pp. 232–247. Springer, 2017. 14
- [DS13] Xin Luna Dong and Divesh Srivastava. “Big data integration.” In *2013 IEEE 29th international conference on data engineering (ICDE)*, pp. 1245–1248. IEEE, 2013. 104, 127
- [DSH17] Xiuquan Du, Shiwei Sun, Changlin Hu, Yu Yao, Yuanting Yan, and Yanping Zhang. “DeepPPI: boosting prediction of protein–protein interactions with deep neural networks.” *Journal of chemical information and modeling*, **57**(6):1499–1510, 2017. 45
- [DSW21] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. “ETA Prediction with Graph Neural Networks in Google Maps.” In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3767–3776, 2021. 4, 125
- [FBI20] FBI. “Crimes in the U.S.” <https://ucr.fbi.gov/crime-in-the-u.s/>, 2020. [Online]. 148
- [FJ18] Police Foundation and U.S. Department of Justice. “Homicide Investigation Case File Profile: The Los Angeles Police Department Murder Book.” <https://www.ojp.gov/ncjrs/virtual-library/abstracts/homicide-investigation-case-file-profile-los-angeles-police>, 2018. [Online]. 151, 157

- [FPS13] Daniel Faria, Catia Pesquita, Emanuel Santos, Matteo Palmonari, Isabel F Cruz, and Francisco M Couto. “The agreementmakerlight ontology matching system.” In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems*”, pp. 527–541. Springer, 2013. [105](#), [126](#)
- [FPW21] Dominic Flocco, Bryce Palmer-Toy, Ruixiao Wang, Hongyu Zhu, Rishi Sonthalia, Junyuan Lin, Andrea L Bertozzi, and P Jeffrey Brantingham. “An Analysis of COVID-19 Knowledge Graph Construction and Applications.” In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 2631–2640. IEEE, 2021. [149](#)
- [FSG18] Antonio Fabregat, Konstantinos Sidiropoulos, Phani Garapati, Marc Gillespie, Kerstin Hausmann, Robin Haw, Bijay Jassal, Steve Jupe, Florian Korninger, Sheldon McKay, Lisa Matthews, Bruce May, Marija Milacic, Karen Rothfels, Veronica Shamovsky, Marissa Webber, Joel Weiser, Mark Williams, Guanming Wu, and Peter D’Eustachio. “The Reactome Pathway Knowledge Base.” *Nucleic acids research*, **46**(D1):D649–D655, 2018. [153](#)
- [FYZ15] Meng Fang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. “Trgraph: Cross-network transfer learning via common signature subgraphs.” *IEEE Transactions on Knowledge and Data Engineering*, **27**(9):2536–2549, 2015. [145](#)
- [FZM20] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. “MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding.” In *WWW*, p. 2331–2341, 2020. [125](#)
- [Gar20] Brandon L. Garrett. “Wrongful Convictions.” *Annual Review of Criminology*, **3**(1):245–259, 2020. [150](#)
- [GB10] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks.” In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010. [52](#), [120](#)

- [GDJ21] Thomas Gaudalet, Ben Day, Arian Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom Blundell, Michael Bronstein, and Jake Taylor-King. “Utilizing graph machine learning within drug discovery and development.” *Briefings in bioinformatics*, **22**(6):bbab159, 2021. 70
- [GG08] Claudio Giuliano and Alfio Gliozzo. “Instance-based ontology population exploiting named-entity substitution.” In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 265–272. Association for Computational Linguistics, 2008. 14
- [GJB20] David Gordon, Gwendolyn Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris White, Matthew O’Meara, Veronica Rezelj, Jeffrey Guo, Danielle Swaney, Tia Tummino, Ruth Hüttenhain, Robyn Kaake, Alicia Richards, Beril Tutuncuoglu, Helene Foussard, Jyoti Batra, Kelsey Haas, Maya Modak, and Nevan Krogan. “A SARS-CoV-2 protein interaction map reveals targets for drug repurposing.” *Nature*, pp. 1–13, 2020. 40, 54, 64, 65
- [GL16] Aditya Grover and Jure Leskovec. “node2vec: Scalable feature learning for networks.” In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864. ACM, 2016. 101, 146
- [GLT20] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. “Retrieval augmented language model pre-training.” In *International Conference on Machine Learning*, pp. 3929–3938. PMLR, 2020. 4
- [GS18] Victor Gutiérrez-Basulto and Steven Schockaert. “From knowledge graph embedding to ontology embedding? An analysis of the compatibility between vector space representations and rules.” In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2018. 14
- [GTC21] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu,

- Tristan Naumann, Jianfeng Gao, and Hoifung Poon. “Domain-specific language model pretraining for biomedical natural language processing.” *ACM Transactions on Computing for Healthcare (HEALTH)*, **3**(1):1–23, 2021. 149
- [GWW16] Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. “Jointly embedding knowledge graphs and logical rules.” In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 192–202, 2016. 14
- [GYW08] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. “Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences.” *Nucleic acids research*, **36**(9):3025–3030, 2008. 45
- [HCT18] Cong Phuoc Huynh, Arridhana Ciptadi, Amrisha Tyagi, and Amit Agrawal. “CRAFT: Complementary Recommendations Using Adversarial Feature Transformer.” *arXiv preprint arXiv:1804.10871*, 2018. 101
- [HCX07] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. “Frequent pattern mining: current status and future directions.” *Data mining and knowledge discovery*, **15**(1):55–86, 2007. 73, 101
- [HCY19] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. “Universal Representation Learning of Knowledge Bases by Jointly Embedding Instances and Ontological Concepts.” In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1709–1719, 2019. 5, 6, 44, 52, 90, 102, 163
- [HDW20a] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. “GPT-GNN: Generative pre-training of graph neural networks.” In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1857–1867, 2020. 4
- [HDW20b] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. “Heterogeneous graph

transformer.” In *Proceedings of The Web Conference 2020*, pp. 2704–2710, 2020.

4

- [HJC20] Junheng Hao, Chelsea J-T Ju, Muhao Chen, Yizhou Sun, Carlo Zaniolo, and Wei Wang. “Bio-JOIE: Joint representation learning of biological knowledge bases.” In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–10, 2020. 4, 7, 146, 149, 163
- [HLE21] Junheng Hao, Chuan Lei, Vasilis Efthymiou, Abdul Quamar, Fatma Özcan, Yizhou Sun, and Wei Wang. “Medto: Medical data to ontology matching using hybrid graph neural networks.” In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2946–2954, 2021. 4, 146, 149, 163
- [HLW18] Lei Huang, Li Liao, and Cathy H Wu. “Completing sparse and disconnected protein-protein network by deep learning.” *BMC bioinformatics*, **19**(1):103, 2018. 41
- [HM16] Ruining He and Julian McAuley. “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.” In *proceedings of the 25th international conference on world wide web*, pp. 507–517. International World Wide Web Conferences Steering Committee, 2016. 77, 89
- [HPM16] Ruining He, Charles Packer, and Julian McAuley. “Learning compatibility across categories for heterogeneous item recommendation.” In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 937–942. IEEE, 2016. 73
- [HSB13] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia.” *Artificial intelligence*, **194**:28–61, 2013. 1

- [HSM15] Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O’Donovan. “The GOA database: gene ontology annotation updates for 2015.” *Nucleic acids research*, **43**(D1):D1057–D1063, 2015. 40
- [HYG15] Yu-An Huang, Zhu-Hong You, Xin Gao, Leon Wong, and Lirong Wang. “Using weighted sparse representation model combined with discrete cosine transformation to predict protein-protein interactions from protein sequence.” *BioMed research international*, **2015**, 2015. 44
- [HYL17a] Will Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs.” In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017. 102, 105, 113, 125, 146
- [HYL17b] William L. Hamilton, Rex Ying, and Jure Leskovec. “Representation Learning on Graphs: Methods and Applications.” *IEEE Data Eng. Bull.*, 2017. 102
- [HZL20] Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. “P-companion: A principled framework for diversified complementary product recommendation.” In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2517–2524, 2020. 1, 4, 7, 125, 146
- [ISM20] Vassilis N. Ioannidis, Xiang Song, Saurav Manchanda, Mufei Li, Xiaoqin Pan, Da Zheng, Xia Ning, Xiangxiang Zeng, and George Karypis. “DRKG - Drug Repurposing Knowledge Graph for Covid-19.” <https://github.com/gnn4dr/DRKG/>, 2020. 67
- [JBZ20] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. “A survey on contrastive self-supervised learning.” *Technologies*, **9**(1):2, 2020. 5

- [JG11] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. “LogMap: Logic-Based and Scalable Ontology Matching.” In *ISWC*, pp. 273–288, 2011. 105, 126
- [JHX15] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. “Knowledge graph embedding via dynamic mapping matrix.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 687–696, 2015. 13
- [JJH21] Jyun-Yu Jiang, Chelsea J-T Ju, Junheng Hao, Muhao Chen, and Wei Wang. “JEDI: circular RNA prediction based on junction encoders and deep interaction among splice sites.” *Bioinformatics*, **37**(Supplement_1):i289–i298, 2021. 4
- [JKH20] Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. “Language Generation with Multi-Hop Reasoning on Commonsense Knowledge Graph.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 725–736, 2020. 4
- [JPS16] Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark. “MIMIC-III, a freely accessible critical care database.” *Scientific data*, **3**:160035, 2016. 105, 118
- [JSM18] Manasa Jammi, Jaydeep Sen, Ashish Mittal, Sagar Verma, Vardaan Pahuja, Rema Ananthanarayanan, Pranay Lohia, Hima Karanam, Diptikalyan Saha, and Karthik Sankaranarayanan. “Tooling framework for instantiating natural language querying system.” *Proceedings of the VLDB Endowment*, **11**(12):2014–2017, 2018. 119, 122
- [JWF16] Katie A. Jacobs, Ashley R. P. Wellman, Amanda M. Fuller, Celeste P. Anderson, and Samantha M. Jurado. “Exploring the familial impact of cold case homicides.” *Journal of Family Studies*, **22**(3):256–271, 2016. 148

- [JWL16] Yantao Jia, Yuanzhuo Wang, Hailun Lin, Xiaolong Jin, and Xueqi Cheng. “Locally adaptive translation for knowledge graph embedding.” In *Thirtieth AAAI conference on artificial intelligence*, 2016. 13
- [KB15a] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization.” *International Conference on Learning Representations (ICLR)*, 2015. 52, 118, 136
- [KB15b] Yehuda Koren and Robert Bell. “Advances in collaborative filtering.” In *Recommender systems handbook*, pp. 77–118. Springer, 2015. 73, 101
- [KBB17] Da Kuang, P. Jeffrey Brantingham, and Andrea L. Bertozzi. “Crime topic modeling.” *Crime Science*, **6**(1):12, 2017. 164
- [KBT15] Denis Krompaß, Stephan Baier, and Volker Tresp. “Type-constrained representation learning in knowledge graphs.” In *International semantic web conference*, pp. 640–655. Springer, 2015. 13
- [KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. “Matrix factorization techniques for recommender systems.” *Computer*, **42**(8):30–37, 08 2009. 73, 101
- [KET19] Takahiro Kawamura, Shusaku Egami, Koutarou Tamura, Yasunori Hokazono, Takanori Ugai, Yusuke Koyanagi, Fumihito Nishino, Seiji Okajima, Katsuhiko Murakami, Kunihiko Takamatsu, Aoi Sugiura, Shun Shiramatsu, Xiangyu Zhang, and Kouji Kozaki. “Report on the first knowledge graph reasoning challenge 2018.” In *Joint International Semantic Technology Conference*, pp. 18–34. Springer, 2019. 149
- [KJM09] Timothy G. Keel, John P. Jarvis, and Yvonne E. Muirhead. “An Exploratory Analysis of Factors Affecting Homicide Investigations:Examining the Dynamics of Murder Clearance Rates.” *Homicide Studies*, **13**(1):50–68, 2009. 148
- [KJR21] Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. “JointGT: Graph-Text Joint Representation Learning for

- Text Generation from Knowledge Graphs.” In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2526–2538, 2021. 4
- [KKK18] Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. “Deepalignment: Unsupervised ontology matching with refined word vectors.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 787–798, 2018. 105, 126
- [KKL19] Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian McAuley. “Complete the Look: Scene-based Complementary Product Recommendation.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10532–10541, 2019. 101
- [KKS18] Prodromos Kolyvakis, Alexandros Kalousis, Barry Smith, and Dimitris Kiritsis. “Biomedical ontology alignment: an approach based on representation learning.” *J. Biomed. Semant.*, **9**(1):21:1–21:20, 2018. 126
- [KSS19] Tjerko Kamminga, Simen-Jan Slagman, Vitor AP Martins dos Santos, Jetta JE Bijlsma, and Peter J Schaap. “Risk-based bioengineering strategies for reliable bacterial vaccine production.” *Trends in biotechnology*, 2019. 40
- [KT12] Alex Kulesza and Ben Taskar. “Determinantal point processes for machine learning.” *Foundations and Trends® in Machine Learning*, **5**(2–3):123–286, 2012. 102
- [KW17] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks.” In *International Conference on Learning Representations (ICLR)*, 2017. 102, 105, 134, 140
- [KWM18] Wang-Cheng Kang, Mengting Wan, and Julian McAuley. “Recommendation Through Mixtures of Heterogeneous Item Relationships.” In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1143–1152. ACM, 2018. 101

- [LAB12] Lydie Lane, Ghislaine Argoud-Puy, Aurore Britan, Isabelle Cusin, Paula Duek, Olivier Evalet, Alain Gateau, Pascale Gaudet, Anne Gleizes, Alexandre Masselot, Catherine Zwahlen, and Amos Bairoch. “neXtProt: a knowledge platform for human proteins.” *Nucleic acids research*, **40**(D1):D76–D83, 2012. 40
- [LCC19] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. “KagNet: Knowledge-Aware Graph Networks for Commonsense Reasoning.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2829–2839, 2019. 4
- [LCT18] Chen Luo, Zhengzhang Chen, Lu-An Tang, Anshumali Shrivastava, Zhichun Li, Haifeng Chen, and Jieping Ye. “TINET: Learning invariant networks via knowledge transfer.” In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1890–1899, 2018. 129, 138, 140, 141, 145
- [Leo15] Jill Leovy. *Ghettoside: A True Story of Murder in America*. Spiegel & Grau, New York, 2015. 148, 150
- [LGD19] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. “Graph matching networks for learning the similarity of graph structured objects.” In *International conference on machine learning*, pp. 3835–3845. PMLR, 2019. 135
- [LGY18] Hang Li, Xiu-Jun Gong, Hua Yu, and Chang Zhou. “Deep neural network based predictions of protein interactions using primary sequences.” *Molecules*, **23**(8):1923, 2018. 45
- [LHL18] Xin Lv, Lei Hou, Juanzi Li, and Zhiyuan Liu. “Differentiating Concepts and Instances for Knowledge Graph Embedding.” In *Proceedings of the 2018 Con-*

- ference on Empirical Methods in Natural Language Processing*, pp. 1971–1979, 2018. 25
- [LHR19] Charles E Loeffler, Jordan Hyatt, and Greg Ridgeway. “Measuring self-reported wrongful convictions among prisoners.” *Journal of Quantitative Criminology*, **35**(2):259–286, 2019. 150
- [LIJ15] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. “DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia.” *Semantic Web*, **6**(2):167–195, 2015. 1, 10, 20, 23, 106, 163
- [LLS15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. “Learning entity and relation embeddings for knowledge graph completion.” In *Twenty-ninth AAAI conference on artificial intelligence*, 2015. 13, 24
- [LM14] Quoc Le and Tomas Mikolov. “Distributed representations of sentences and documents.” In *International conference on machine learning*, pp. 1188–1196. PMLR, 2014. 159
- [LNK19] Qi Liu, Maximilian Nickel, and Douwe Kiela. “Hyperbolic graph neural networks.” *Advances in Neural Information Processing Systems*, **32**, 2019. 126
- [LNM17] Chia Pui Ling, Noor Maizura Mohamad Noor, and Fatihah Mohd. “Knowledge representation model for crime analysis.” *Procedia computer science*, **116**:484–491, 2017. 4
- [LOQ18] Chuan Lei, Fatma Özcan, Abdul Quamar, Ashish R Mittal, Jaydeep Sen, Diptikalyan Saha, and Karthik Sankaranarayanan. “Ontology-Based Natural Language Query Interfaces for Data Exploration.” *IEEE Data Engineering*, **41**(3):52–63, 2018. 106, 111, 119, 122

- [LSY03] Greg Linden, Brent Smith, and Jeremy York. “Amazon. com recommendations: Item-to-item collaborative filtering.” *IEEE Internet computing*, **7**(1):76–80, 2003. [73](#), [101](#)
- [LWH21] Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. “Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6418–6425, 2021. [4](#)
- [LYF21] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.” *arXiv preprint arXiv:2107.13586*, 2021. [5](#)
- [LYK20] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining.” *Bioinformatics*, **36**(4):1234–1240, 2020. [114](#), [154](#), [155](#), [163](#)
- [LZW17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. “Deep transfer learning with joint adaptation networks.” In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017. [145](#)
- [LZZ20] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. “K-bert: Enabling language representation with knowledge graph.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2901–2908, 2020. [4](#)
- [MAG14] Hamid Mousavi, Maurizio Atzori, Shi Gao, and Carlo Zaniolo. “Text-mining, structured queries, and knowledge management on web document corpora.” *ACM SIGMOD Record*, **43**(3):48–54, 2014. [14](#)

- [MBS14] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. “Yago3: A knowledge base from multilingual wikipe-dias.” In *7th biennial conference on innovative data systems research*. CIDR Conference, 2014. 1, 10, 20, 23
- [MCL22] Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. “Open Domain Question Answering with A Unified Knowledge Interface.” In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1605–1620, 2022. 4
- [MCW18] Jianxin Ma, Peng Cui, Xiao Wang, and Wenwu Zhu. “Hierarchical taxonomy aware network embedding.” In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1920–1929. ACM, 2018. 13, 52
- [MDJ17] Shiheng Ma, Jianhui Ding, Weijia Jia, Kun Wang, and Minyi Guo. “Transt: Type-based multiple embedding representations for knowledge graph completion.” In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 717–733. Springer, 2017. 13
- [MFT20] Lauren A. Magee, J. Dennis Fortenberry, Wanzhu Tu, and Sarah E. Wiehe. “Neighborhood variation in unsolved homicides: a retrospective cohort study in Indianapolis, Indiana, 2007–2017.” *Injury Epidemiology*, 7(1):61, 2020. 148
- [MHR19] Stavros Makrodimitris, Roeland C.H.J. van Ham, and Marcel J.T. Reinders. “Sparsity of Protein-Protein Interaction Networks Hinders Function Prediction in Non-Model Species.” *bioRxiv*, 2019. 41
- [Mil95] George A Miller. “WordNet: a lexical database for English.” *Communications of the ACM*, 38(11):39–41, 1995. 1
- [MMR08] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. “Domain adaptation with multiple sources.” *Advances in neural information processing systems*, 21, 2008. 145

- [MPL15] Julian McAuley, Rahul Pandey, and Jure Leskovec. “Inferring networks of substitutable and complementary products.” In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2015. 1, 74, 89, 90, 101
- [MPR05] Shie Mannor, Dori Peleg, and Reuven Rubinfeld. “The cross entropy method for classification.” In *Proceedings of the 22nd international conference on Machine learning*, pp. 561–568, 2005. 157
- [MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality.” *Advances in neural information processing systems*, **26**, 2013. 44, 55
- [MTS15] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. “Image-based recommendations on styles and substitutes.” In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. ACM, 2015. 77, 89
- [NGP21] Hoang-Van Nguyen, Francesco Gelli, and Soujanya Poria. “DOZEN: cross-domain zero shot named entity recognition with knowledge graph.” In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 1642–1646, 2021. 4
- [NK17] Maximilian Nickel and Douwe Kiela. “Poincaré embeddings for learning hierarchical representations.” *Advances in neural information processing systems*, **30**, 2017. 4, 114, 126
- [NRP16] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. “Holographic embeddings of knowledge graphs.” In *Thirtieth Aaai conference on artificial intelligence*, 2016. 13, 19, 22, 24, 47
- [NTK11] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. “A three-way model for collective learning on multi-relational data.” In *Proceedings of the 28th In-*

- ternational Conference on International Conference on Machine Learning*, pp. 809–816. Omnipress, 2011. 13
- [NYH16] Tanachat Nilanon, Jiayu Yao, Junheng Hao, Sanjay Purushotham, and Yan Liu. “Normal/abnormal heart sound recordings classification using convolutional neural network.” In *2016 computing in cardiology conference (CinC)*, pp. 585–588. IEEE, 2016. 4
- [OKK18] Naoki Otani, Hirokazu Kiyomaru, Daisuke Kawahara, and Sadao Kurohashi. “Cross-lingual Knowledge Projection Using Machine Translation and Target-side Knowledge Base Completion.” In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1508–1520, 2018. 14
- [OSB19] Rose Oughtred, Chris Stark, Bobby-Joe Breitzkreutz, Jennifer Rust, Lorie Boucher, Christie Chang, Nadine Kolas, Lara O’Donnell, Genie Leung, Rochelle McAdam, Frederick Zhang, Sonam Dolma, Andrew Willems, Jasmin Coulombe-Huntington, Andrew Chatr-Aryamontri, Kara Dolinski, and Mike Tyers. “The BioGRID interaction database: 2019 update.” *Nucleic acids research*, **47**(D1):D529–D541, 2019. 41
- [PAG17] Jay Pujara, Eriq Augustine, and Lise Getoor. “Sparsity and noise: Where knowledge graph embeddings fall short.” In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1751–1756, 2017. 27, 29
- [PAS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “Deepwalk: Online learning of social representations.” In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710. ACM, 2014. 101
- [PBU20] Ritika Pandey, P Jeffrey Brantingham, Craig D Uchida, and George Mohler. “Building knowledge graphs of homicide investigation chronologies.” In *2020*

- International Conference on Data Mining Workshops (ICDMW)*, pp. 790–798. IEEE, 2020. [4](#), [148](#), [149](#), [153](#), [159](#), [164](#)
- [PDC20] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. “Evolvegen: Evolving graph convolutional networks for dynamic graphs.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5363–5370, 2020. [108](#)
- [PGM19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Közcanpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and Soumith Chintala. “Pytorch: An imperative style, high-performance deep learning library.” *Advances in neural information processing systems*, **32**, 2019. [155](#)
- [PLK21] Heather Prince, Cynthia Lum, and Christopher S Koper. “Effective police investigative practices: an evidence-assessment of the research.” *Policing: An International Journal*, 2021. [148](#)
- [PMM20] Irene Papatheodorou, Pablo Moreno, Jonathan Manning, Alfonso Fuentes, Nancy George, Silvie Fexova, Nuno Fonseca, Anja Füllgrabe, Matthew Green, Ni Huang, Laura Huerta, Haider Iqbal, Monica Jianu, Suhaib Mohammed, Lingyun Zhao, Andrew Jarnuczak, Simon Jupp, John Marioni, Kerstin Meyer, and Alvis Brazma. “Expression Atlas update: from tissues to single cells.” *Nucleic acids research*, **48**(D1):D77–D83, 2020. [40](#)
- [PNI18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. “Deep contextualized word representations.” *CoRR*, **abs/1802.05365**, 2018. [113](#)
- [POK16] Oliver Philipp, Heinz D Osiewacz, and Ina Koch. “Path2PPI: an R package to

- predict protein–protein interaction networks for a set of proteins.” *Bioinformatics*, **32**(9):1427–1429, 2016. 45
- [PR13] Jeff Pasternack and Dan Roth. “Latent credibility analysis.” In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1009–1020. ACM, 2013. 14
- [PRR19] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. “Language Models as Knowledge Bases?” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019. 4
- [PWS20] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. “Yago 4: A reason-able knowledge base.” In *European Semantic Web Conference*, pp. 583–596. Springer, 2020. 1, 163
- [PYL19] Yifan Peng, Shankai Yan, and Zhiyong Lu. “Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets.” In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pp. 58–65, 2019. 163
- [QCZ14] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. “Contextual combinatorial bandit and its application on diversified online recommendation.” In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 461–469. SIAM, 2014. 102
- [QDM18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. “Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec.” In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 459–467, 2018. 5, 146

- [QLM20] Abdul Quamar, Chuan Lei, Dorian Miller, Fatma Ozcan, Jeffrey Kreulen, Robert J Moore, and Vasilis Efthymiou. “An ontology-based conversation system for knowledge bases.” In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 361–376, 2020. 104
- [Q SX20] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. “Pre-trained models for natural language processing: A survey.” *Science China Technological Sciences*, **63**(10):1872–1897, 2020. 149, 163
- [QW17] Nadeem Qazi and B. L. William Wong. “Behavioural and Tempo-Spatial Knowledge Graph for Crime Matching through Graph Theory.” In *2017 European Intelligence and Security Informatics Conference (EISIC)*, pp. 143–146, 2017. 164
- [RCO19] Rebecca Riggs, Richard Timothy Coupe, and Denis O’Connor. *Homicide resources, solvability and detection*, pp. 331–365. Springer, 2019. 148
- [RG19] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. 159
- [RJM20] Wendy C Regoeczi, John Jarvis, and Ashley Mancik. “Homicide investigations in context: Exploring explanations for the divergent impacts of victim race, gender, elderly victims, and firearms on homicide clearances.” *Homicide Studies*, **24**(1):25–44, 2020. 166
- [RKK18] Sashank Reddi, Satyen Kale, and Sanjiv Kumar. “On the convergence of Adam and Beyond.” In *International Conference on Learning Representations*, 2018. 21

- [RM03] Cornelius Rosse and José LV Mejino Jr. “A reference ontology for biomedical informatics: the Foundational Model of Anatomy.” *Journal of biomedical informatics*, **36**(6):478–500, 2003. [119](#)
- [Rot17] Jeffrey Roth. “A city-level analysis of property crime clearance rates.” *Criminal Justice Studies*, **30**(1):45–62, 2017. [148](#)
- [RSR15] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. “Injecting logical background knowledge into embeddings for relation extraction.” In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1119–1129, 2015. [14](#)
- [RWC19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. “Language models are unsupervised multitask learners.” *OpenAI blog*, **1**(8):9, 2019. [159](#)
- [RWC20] Cynthia Rudin, Caroline Wang, and Beau Coker. “The Age of Secrecy and Unfairness in Recidivism Prediction.” *Harvard Data Science Review*, **2**(1), 2020. [167](#)
- [RWE15] Ian Robinson, Jim Webber, and Emil Eifrem. *Graph databases: new opportunities for connected data.* ” O’Reilly Media, Inc.”, 2015. [151](#)
- [RZP22] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando Freitas. “A generalist agent.” *arXiv preprint arXiv:2205.06175*, 2022. [172](#)
- [SCH17] Robert Speer, Joshua Chin, and Catherine Havasi. “Conceptnet 5.5: An open multilingual graph of general knowledge.” In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. [1](#), [10](#), [24](#), [153](#), [154](#)

- [SCM13] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. “Reasoning with neural tensor networks for knowledge base completion.” In *Advances in neural information processing systems*, pp. 926–934, 2013. 13
- [SDN19] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. “RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space.” In *International Conference on Learning Representations*, 2019. 102
- [SG08] Ajit P Singh and Geoffrey J Gordon. “Relational learning via collective matrix factorization.” In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 650–658, 2008. 146
- [SGH18] Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. “Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations.” *Bioinformatics*, **34**(13):i52–i60, 2018. 42, 44, 55
- [SH12] Yizhou Sun and Jiawei Han. “Mining heterogeneous information networks: principles and methodologies.” *Synthesis Lectures on Data Mining and Knowledge Discovery*, **3**(2):1–159, 2012. 3
- [SHL17] Zequn Sun, Wei Hu, and Chengkai Li. “Cross-lingual entity alignment via joint attribute-preserving embedding.” In *International Semantic Web Conference*, pp. 628–644. Springer, 2017. 14, 44
- [SHP16] Chenshuo Sun, Junheng Hao, Xin Pei, Zuo Zhang, and Yi Zhang. “A data-driven approach for duration evaluation of accident impacts on urban intersection traffic flow.” In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1354–1359. IEEE, 2016. 4
- [SHY11] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks.” *Proceedings of the VLDB Endowment*, **4**(11):992–1003, 2011. 3, 146

- [SHZ18] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. “Bootstrapping entity alignment with knowledge graph embedding.” In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4396–4402. AAAI Press, 2018. [14](#), [112](#)
- [SK09] Xiaoyuan Su and Taghi M Khoshgoftaar. “A survey of collaborative filtering techniques.” *Advances in artificial intelligence*, **2009**, 2009. [73](#), [101](#)
- [SK21] Tara Safavi and Danai Koutra. “Relational World Knowledge Representation in Contextual Language Models: A Review.” In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1053–1067, 2021. [4](#)
- [SKB18] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. “Modeling relational data with graph convolutional networks.” In *European Semantic Web Conference*, pp. 593–607. Springer, 2018. [4](#), [102](#), [108](#), [116](#), [125](#), [134](#), [140](#)
- [SKK01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. “Item-based collaborative filtering recommendation algorithms.” In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001. [73](#), [101](#)
- [SKW07] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. “Yago: a core of semantic knowledge.” In *Proceedings of the 16th international conference on World Wide Web*, pp. 697–706, 2007. [1](#), [106](#)
- [SMC16] Damian Szklarczyk, John Morris, Helen Cook, Michael Kuhn, Stefan Wyder, Milan Simonovic, Alberto Santos, Nadezhda Doncheva, Alexander Roth, Peer Bork, Lars Jensen, and Christian von Mering. “The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible.” *Nucleic acids research*, p. gkw937, 2016. [1](#), [40](#), [53](#)
- [SMG14] Andrew M Saxe, James L McClelland, and Surya Ganguli. “Exact solutions to

- the nonlinear dynamics of learning in deep linear neural networks.” *International Conference on Learning Representations*, 2014. 21, 52
- [SPH18] Chenshuo Sun, Xin Pei, Junheng Hao, Yewen Wang, Zuo Zhang, and SC Wong. “Role of road network features in the evaluation of incident impacts on urban traffic mobility.” *Transportation research part B: methodological*, **117**:101–116, 2018. 4
- [SRG16] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning.” *IEEE transactions on medical imaging*, **35**(5):1285–1298, 2016. 145
- [Sri18] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd, 2018. 153
- [ST19] K Srinivasa and P. Santhi Thilagam. “Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers.” *Information Processing & Management*, **56**(6):102059, 2019. 164
- [SWF21] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, and Haifeng Wang. “Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.” *arXiv preprint arXiv:2107.02137*, 2021. 171
- [SWH20] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. “Knowledge graph alignment network with gated multi-hop

- neighborhood aggregation.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 222–229, 2020. 105, 127
- [SZH20] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. “A Benchmarking Study of Embedding-based Entity Alignment for Knowledge Graphs.” *Proceedings of the VLDB Endowment*, **13**(11):2326–2340, 2020. 4, 126, 127, 132
- [SZL17] Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. “Sequence-based prediction of protein protein interaction using a deep-learning algorithm.” *BMC bioinformatics*, **18**(1):277, 2017. 45
- [TBC21] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. “Understanding the capabilities, limitations, and societal impact of large language models.” *arXiv preprint arXiv:2102.02503*, 2021. 165
- [TQZ19] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. “Entity alignment between knowledge graphs using attribute embeddings.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 297–304, 2019. 44
- [TSD18] Rakshit Trivedi, Bunyamin Sisman, Xin Luna Dong, Christos Faloutsos, Jun Ma, and Hongyuan Zha. “LinkNBed: Multi-Graph Representation Learning with Entity Linkage.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 252–262, 2018. 132
- [TSW09] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. “Social influence analysis in large-scale networks.” In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 807–816, 2009. 137
- [TWM12] Paul D Thomas, Valerie Wood, Christopher J Mungall, Suzanna E Lewis, Judith A Blake, Gene Ontology Consortium, et al. “On the use of gene ontology annotations to assess functional similarity among orthologs and paralogs: a short report.” *PLoS computational biology*, **8**(2), 2012. 41

- [TWR16] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. “Complex embeddings for simple link prediction.” In *International conference on machine learning*, pp. 2071–2080. PMLR, 2016. 13, 102
- [Vau20] Paige E Vaughn. “The effects of devaluation and solvability on crime clearance.” *Journal of Criminal Justice*, **68**:101657, 2020. 150, 166
- [VCC18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. “Graph Attention Networks.” *International Conference on Learning Representations*, 2018. 81, 83, 102, 125, 140, 163
- [VK14] Denny Vrandečić and Markus Krötzsch. “Wikidata: a free collaborative knowledgebase.” *Communications of the ACM*, **57**(10):78–85, 2014. 1, 153, 154
- [VSN19] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. “Composition-based Multi-Relational Graph Convolutional Networks.” In *International Conference on Learning Representations*, 2019. 4, 146
- [VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need.” *Advances in neural information processing systems*, **30**, 2017. 4, 117, 154, 163
- [WDR21] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian Suchanek. “Machine knowledge: Creation and curation of comprehensive knowledge bases.” *Foundations and Trends® in Databases*, **10**(2-4):108–490, 2021. 4, 163
- [WDS19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Scao, Sylvain Gugger, and Alexander Rush. “Huggingface’s transformers: State-of-the-art natural language processing.” *arXiv preprint arXiv:1910.03771*, 2019. 155

- [WHC19] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. “Kgat: Knowledge graph attention network for recommendation.” In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 950–958, 2019. 140, 163
- [WJR18] Zihan Wang, Ziheng Jiang, Zhaochun Ren, Jiliang Tang, and Dawei Yin. “A path-constrained framework for discriminating substitutable and complementary products in e-commerce.” In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 619–627. ACM, 2018. 74, 90, 101
- [WJS19] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. “Heterogeneous graph attention network.” In *The world wide web conference*, pp. 2022–2032, 2019. 108, 125
- [WKN20] Zuozhi Wang, Avinash Kumar, Shengquan Ni, and Chen Li. “Demonstration of interactive runtime debugging of distributed dataflows in Texera.” *Proceedings of the VLDB Endowment*, **13**(12):2953–2956, 2020. 67
- [WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. “A survey of transfer learning.” *Journal of Big data*, **3**(1):1–40, 2016. 145
- [WLB06] Ting Wang, Yaoyong Li, Kalina Bontcheva, Hamish Cunningham, and Ji Wang. “Automatic extraction of hierarchical relations from text.” In *European Semantic Web Conference*, pp. 215–229. Springer, 2006. 14
- [WLF19] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. “Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs.” In *IJCAI*, pp. 5278–5284, 2019. 105, 120, 127
- [WLL18] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. “Cross-lingual knowledge graph alignment via graph convolutional networks.” In *Proceedings of the*

- 2018 conference on empirical methods in natural language processing*, pp. 349–357, 2018. [105](#), [120](#), [126](#)
- [WLZ16] Yue Wu, Jingfei Li, Peng Zhang, and Dawei Song. “Learning to improve affinity ranking for diversity search.” In *Asia Information Retrieval Symposium*, pp. 335–341. Springer, 2016. [102](#)
- [WMG19] Romain Warlop, Jérémie Mary, and Mike Gartrell. “Tensorized Determinantal Point Processes for Recommendation.” In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1605–1615. ACM, 2019. [102](#)
- [WMR21] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Hendricks, and Iason Gabriel. “Ethical and social risks of harm from Language Models.” *arXiv preprint arXiv:2112.04359*, 2021. [167](#)
- [WMW17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. “Knowledge graph embedding: A survey of approaches and applications.” *IEEE Transactions on Knowledge and Data Engineering*, **29**(12):2724–2743, 2017. [4](#), [5](#), [13](#), [102](#), [141](#), [149](#), [163](#)
- [Wor10] Richard Wortley. *Critiques of situational crime prevention*, p. 884–886. Sage, Thousand Oaks, CA, 2010. [164](#)
- [WPC20] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. “A comprehensive survey on graph neural networks.” *IEEE transactions on neural networks and learning systems*, **32**(1):4–24, 2020. [146](#)
- [WPZ20] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. “Unsu-

- pervised domain adaptive graph convolutional networks.” In *Proceedings of The Web Conference 2020*, pp. 1457–1467, 2020. [140](#), [145](#)
- [WSZ20] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. “Graph neural networks in recommender systems: a survey.” *ACM Computing Surveys (CSUR)*, 2020. [4](#)
- [WW08] Fei Wu and Daniel S Weld. “Automatically refining the wikipedia infobox ontology.” In *Proceedings of the 17th international conference on World Wide Web*, pp. 635–644, 2008. [1](#)
- [WYL15] Leon Wong, Zhu-Hong You, Shuai Li, Yu-An Huang, and Gang Liu. “Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel PR-LPQ descriptor.” In *International Conference on Intelligent Computing*, pp. 713–720. Springer, 2015. [45](#)
- [WYL17] Yan-Bin Wang, Zhu-Hong You, Xiao Li, Tong-Hai Jiang, Xing Chen, Xi Zhou, and Lei Wang. “Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network.” *Molecular BioSystems*, **13**(7):1336–1344, 2017. [45](#)
- [WZF14] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. “Knowledge graph embedding by translating on hyperplanes.” In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014. [13](#), [22](#), [44](#), [48](#)
- [XCK18] Guohui Xiao, Diego Calvanese, Roman Kontchakov, Domenico Lembo, Antonella Poggi, Riccardo Rosati, and Michael Zakharyashev. “Ontology-based data access: a survey.” In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 5511–5519, 2018. [104](#)
- [XLS16] Ruobing Xie, Zhiyuan Liu, and Maosong Sun. “Representation learning of knowledge graphs with hierarchical types.” In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2965–2971, 2016. [13](#)

- [YCH15] Zhu-Hong You, Keith CC Chan, and Pengwei Hu. “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), 2015. 44, 45
- [YCZ13] Jihang Ye, Hong Cheng, Zhe Zhu, and Minghua Chen. “Predicting positive and negative links in signed social networks by transfer learning.” In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1477–1488, 2013. 146
- [YDC22] Da Yin, Li Dong, Hao Cheng, Xiaodong Liu, Kai-Wei Chang, Furu Wei, and Jianfeng Gao. “A survey of knowledge-intensive nlp with pre-trained language models.” *arXiv preprint arXiv:2202.08772*, 2022. 4
- [YHC18] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. “Graph convolutional neural networks for web-scale recommender systems.” In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983. ACM, 2018. 102
- [YHZ22] Yunzhi Yao, Shaohan Huang, Ningyu Zhang, Li Dong, Furu Wei, and Huajun Chen. “Kformer: Knowledge injection in transformer feed-forward layers.” *arXiv preprint arXiv:2201.05742*, 2022. 4
- [YRB21] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. “QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering.” In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 535–546, 2021. 4
- [YTY21] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. “Ernie-vil: Knowledge enhanced vision-language representations through

- scene graphs.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3208–3216, 2021. 4, 172
- [YWC18] Jinyoung Yeo, Geungyu Wang, Hyunsouk Cho, Seungtaek Choi, and Seungwon Hwang. “Machine-Translated Knowledge Transfer for Commonsense Causal Reasoning.” In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 14
- [YYH15] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. “Embedding Entities and Relations for Learning and Inference in Knowledge Bases.” In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015. 13, 19, 22, 24, 25, 36, 47, 52, 102, 136
- [YYM18] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. “Hierarchical graph representation learning with differentiable pooling.” *Advances in neural information processing systems*, 31, 2018. 113
- [YZL22] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. “A survey of knowledge-enhanced text generation.” *ACM Computing Surveys (CSUR)*, 2022. 4
- [YZQ22] Wenhao Yu, Chenguang Zhu, Lianhui Qin, Zhihan Zhang, Tong Zhao, and Meng Jiang. “Diversifying Content Generation for Commonsense Reasoning with Mixture of Knowledge Graph Experts.” In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1896–1906, 2022. 4
- [YZZ14] Zhu-Hong You, Lin Zhu, Chun-Hou Zheng, Hong-Jie Yu, Su-Ping Deng, and Zhen Ji. “Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set.” In *BMC bioinformatics*, volume 15, p. S9. Springer, 2014. 45
- [ZAL18] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. “Modeling polypharmacy

- side effects with graph convolutional networks.” *Bioinformatics*, **34**(13):i457–i466, 2018. 40
- [ZBY22] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. “GreaseLM: Graph REASoning Enhanced Language Models for Question Answering.” *arXiv preprint arXiv:2201.08860*, 2022. 4
- [ZHL14] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. “Bundle recommendation in ecommerce.” In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 657–666. ACM, 2014. 101
- [ZHS20] Yadi Zhou, Yuan Hou, Jiayu Shen, Yin Huang, William Martin, and Feixiong Cheng. “Network-based drug repurposing for novel coronavirus 2019-nCoV/SARS-CoV-2.” *Cell discovery*, **6**(1):1–18, 2020. 40
- [ZLN18] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. “Quality-aware neural complementary item recommendation.” In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 77–85. ACM, 2018. 101
- [ZLW22] Nansu Zong, Ning Li, Andrew Wen, Victoria Ngo, Yue Yu, Ming Huang, Shaika Chowdhury, Chao Jiang, Sunyang Fu, Richard Weinshilboum, Guoqian Jiang, Lawrence Hunter, and Hongfang Liu. “BETA: a comprehensive benchmark for computational drug-target prediction.” *Briefings in bioinformatics*, p. bbac199, 2022. xiii, 45, 70
- [ZSH19] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. “Multi-view Knowledge Graph Embedding for Entity Alignment.” In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 5429–5435, 08 2019. 44

- [ZSW22] Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, Junheng Hao, Ye-Yi Wang, Kuansan Wang, and Jiawei Han. “Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification.” In *Proceedings of the ACM Web Conference 2022*, pp. 3162–3173, 2022. 4, 163
- [ZWN09] Jiaqian Zheng, Xiaoyuan Wu, Junyu Niu, and Alvaro Bolivar. “Substitutes or complements: another step forward in recommendations.” In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 139–146. ACM, 2009. 101
- [ZXL17] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. “Iterative entity alignment via joint knowledge embeddings.” In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4258–4264. AAAI Press, 2017. 14
- [ZXR22] Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Lin, Meng Jiang, and Wenhao Yu. “Knowledge-Augmented Methods for Natural Language Processing.” In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 12–20, 2022. 171
- [ZY21] Yu Zhang and Qiang Yang. “A survey on multi-task learning.” *IEEE Transactions on Knowledge and Data Engineering*, 2021. 146
- [ZYS19] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. “Deep learning based recommender system: A survey and new perspectives.” *ACM Computing Surveys (CSUR)*, **52**(1):5, 2019. 101