Multiple Objective Learning for Effective Knowledge Graph Embedding

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Afshin Sadeghi
Geboren in
Fasa, Iran

Bonn 2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Jens Lehmann
2. Gutachter: Prof. Dr. Stefan Wrobel

Tag der Promotion: 17.02.2023
Erscheinungsjahr: 2023

Abstract

Over the past decade, knowledge graphs (KGs) have become popular for capturing structured domain knowledge. Knowledge graphs particularly allow the effortless integration of heterogeneous data into a coherent model. Besides applications in data integration, KGs are at the center of many artificial intelligence studies as an expressive data model for representation learning purposes. Knowledge graph embedding (KGE) methods produce a latent representation of KGs entities showing applicability potential for solving downstream tasks, including link prediction and node classification. KGE also supports Named Entity Resolution in NLP tasks and is applied in Question Answering Systems. Existing KGE models have achieved excellent results over simple knowledge graphs, where they contain only a few relation patterns that leak into each other. Besides, in simple knowledge graphs, the amount of entities with similar neighbors is lower, and the structure of the subgraphs is unique, so their entities are more easily distinguished. This work dives more into the study of KGE for complex knowledge graphs. In such KGs, distinct relation patterns show up significantly more, and similar substructures repeat over the network on a larger scale. Therefore, recognizing unique entities with limited knowledge about the direct neighbors and the limited recognition of relation patterns is remarkably more difficult. Complex knowledge graph embedding provides several challenges, such as understanding learning distinct relation patterns and graphical features of nodes. The lack of suitable datasets that emulate the difficulty of more complex knowledge graphs further adds to research gaps. Hence, in this thesis, we focus on the research objective of laying the foundations for the advancement of the state-of-the-art to better embed complex knowledge graphs by providing techniques to solve various challenges and resources to fill the research gaps.

First, to effectively target the complex KGE challenge, we propose a multi-objective method that allows learning several relation patterns of knowledge graphs. Multiple Distance Embedding (MDE) generalizes over several distance-based models and proposes combined learning objectives that extract more knowledge from KGs as the base training data. Our solution is based on the principle that we can collectively train and predict using contradicting distance terms by learning independent embedding vectors for each of the terms. We demonstrate that MDE allows modeling relations with (anti)symmetry, inversion, and composition patterns. Our empirical investigation shows the on-par and better performance of MDE relative to the state-of-the-art methods in the link prediction task and its effectiveness in learning individual relation patterns.

We then propose a novel KGE method named Graph Feature Attentive Neural Network (GFA-NN) that computes graphical features of entities. This method addresses the limitation of embedding models that consider only the local graph structure related to an entity and have difficulty distinguishing similar graph substructures. Consequently, the resulting embeddings of GFA-NN are attentive to two types of global network features. First, nodes' relative centrality is based on the observation that some entities are more "prominent" than others. The second is the relative position of entities in the graph. GFA-NN computes several centrality values per entity, generates a random set of reference entities, and computes a given entity's shortest path to each entity in the reference set. It then learns this information by optimizing the objectives specified on each feature. We investigate GFA-NN on several link prediction benchmarks in the inductive and transductive settings. We demonstrate that GFA-NN improves the state-of-the-art records of KGE solutions, especially on large-scale and more complex knowledge graphs. We next construct 96 datasets replicating 16 different relation pattern circumstances and four different inductive and transductive test settings. This effort addresses the gap of missing a leak-free link prediction benchmark of embedding methods on complex knowledge graphs. Our analysis of state-of-the-art models over these datasets provides a better insight into the suitable parameters for each situation, optimizing the KG-embedding-based systems. In addition, to provide reproducible KGE Benchmarking, we create an evaluation Framework based on dockerized test environments. We further developed an application component as the last step of an End-to-End machine learning workflow to overcome the challenges of real-world Link Prediction using knowledge graph embedding. The proposed approaches mentioned in this thesis and the published resources are available at https://github.com/afshinsadeghi/GFA-NN and in https://github.com/mlwin-de/ are released under the umbrella MLwin project.

Acknowledgements

Throughout my Ph.D. studies, several individuals and groups have provided support and direction, without which this work would not have been feasible. Prof. Dr. Jens Lehmann, my supervisor, is to be thanked for allowing me to work on a thesis under his supervision. He gave me confidence and a research vision. He pushed me to carry out my research moreover to fulfill such faith when I presented my study interests, and he decided to back me with faith and spurred me to consider the challenging research targets and top venues. I was extremely fortunate to have him as my supervisor. His belief in my potential and ability, as well as his great leadership, have always motivated me to thrive in my studies and, over time, to take on more responsibilities. Even though he had to relocate to Dresden during the last years of my studies, I appreciate his continued attention and support in my work.

Similarly, I am thankful to Prof. Dr. Sören Auer and Prof. Dr. Stefan Wrobel for their support provided during the course of this thesis.

Dr. Diego Collarana, Dr. Damien Graux, Dr. Christoph Lange, and Dr. María-Esther Vidal have been my friends, coauthors, and research mentors. I am genuinely grateful for their support that goes beyond academia into friendship, and I am happy to have interacted with their research approach. I thank Prof. Dr. Marvin Minsky for supporting my passion in the study of machine learning towards human-centered Artificial Intelligence by introducing me to discussion groups on this topic. These passionate discussions later influenced my derivation to study subjects in Semantic Web and machine learning on Knowledge Graphs.

I am finally very thankful to my family, that has supported me all along in different stages of my life; no matter how near or far, they gave me their bests any time. I would like to express my gratitude to my colleagues, Dr. Najmehsadat Mousavinezhad, Dr. Isaiah Onando Mulang, Dr. Fathoni A. Musyaffa, Dr. Elisa Margareth Sibarani, Dr. Mirette Elias, Firas Kassawat, Farshad Bakhshandegan, Dr. Denis Lukovnikov, Dr. Gëzim Sejdiu, Dr. Mohnish Dubey and Dr. Steffen Lohmann for motivating me by their passion and encouraging me to achieve better results.

Contents

1 Introduction 1
1.1 Motivation 4
1.2 Problem Statement and Challenges 5
1.3 Research Questions 9
1.4 Thesis Overview 11
1.4.1 Contributions 12
1.4.2 Publications 15
1.5 Thesis Structure 16

2 Preliminaries 17
2.1 Knowledge Representation in Networks 17
2.2 Learning Representations for Knowledge Graphs 18
2.3 Relational Pattern Learning 20
2.4 Normalisation and Normal Form 21
2.5 Optimization methods for KGE 22
2.6 Graphical Feature of Knowledge Graphs 24

3 Related Work 25
3.1 Knowledge Graph Embedding: Modeling Methods 25
3.2 Advances in Knowledge Graph Elements 28
3.3 Link Prediction Benchmarking Datasets 29

4 Multiple Distance Embeddings for Link Prediction in Knowledge Graphs 31
4.1 Relation Pattern Coverage in KGE models 33
4.2 MDE: Multiple Distance Embeddings 34
4.3 Model Extensions 38
4.3.1 Searching for the limits in the limit-based Loss 38
4.3.2 MDE$_{NN}$: MDE as a Neural Network 40
4.4 Experiments 41
4.4.1 Relation Pattern Implicit Inference 44
4.4.2 Link Prediction Results 45
4.4.3 Ablation Study 47
4.5 Conclusion 48

5 Graph Feature Aware Knowledge Graph Embedding 49
5.1 Background 52

5.2 Method . . . 53
5.2.1 Model Formulation . . . 54
5.2.2 Centrality-Attentive embedding: . . . 55
5.2.3 Position-Attentive embedding: . . . 56
5.3 Theoretical analysis . . . 57
5.3.1 Connection to Preceding KGE Methods . . . 57
5.3.2 Expressive Power . . . 58
5.3.3 Complexity Analysis . . . 59
5.4 Experiments . . . 59
5.4.1 Transductive link prediction experiment . . . 60
5.4.2 Inductive link prediction experiment . . . 61
5.5 Conclusion . . . 62

6 BenchEmbed: A Framework for Reproducible Link Prediction Benchmarking 63
6.1 Structure: . . . 64
6.2 Functionalities . . . 65
6.3 Demonstration . . . 66
6.4 Defining new Benchmark Environments: . . . 66
6.5 Conclusion . . . 68

7 Relational Pattern Benchmarking 69
7.1 Integration into BenchEmbedd Benchmarking Platform . . . 71
7.2 Related Work . . . 72
7.2.1 Benchmarking . . . 72
7.2.2 Dataset Building . . . 72
7.3 Pattern extraction from Relational Datasets . . . 73
7.3.1 Symmetry . . . 73
7.3.2 Anti-Symmetry . . . 74
7.3.3 Inverse . . . 74
7.3.4 Composition . . . 74
7.3.5 Inference . . . 74
7.4 Evaluation Dataset Settings . . . 75
7.5 A New Set of Pattern Specific Datasets . . . 75
7.5.1 The Standard Patterns . . . 76
7.6 Results . . . 78
7.6.1 Discussion . . . 81
7.6.2 Evaluation of GFA-NN model . . . 82
7.6.3 Comparison to Aggregate datasets . . . 83
7.7 Conclusion . . . 83

8 Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter 85
8.1 Background . . . 87
8.2 Related Work . . . 88
8.3 KG Embeddings for Twitter Link Prediction . . . 88

8.4 Experiments . . . 90
8.4.1 Performance Evaluation . . . 90
8.4.2 Link Prediction Analysis: . . . 91
8.5 Discussion of the specificity of the problem . . . 92
8.6 Conclusion . . . 92

9 Conclusion and Future Directions 93
9.1 Review of the Contributions . . . 94
9.2 Limitations and Future Directions . . . 98
9.3 Closing Remarks . . . 100

Bibliography 101

A List of Publications 117

List of Figures 119

List of Tables 121

CHAPTER 1

Introduction

> Most of these difficulties disappear when we consider one-one immersions $f : P \to M$ which are homeomorphisms onto their image. Such an immersion is called **imbedding** (**embedding** for the English).
>
> [1], p. 49, The suggestion of the term "embedding" to use in the mathematical context by Michael Spivak.

Nature offers us data often in the form of a graph. The nesting of the human population on earth is in a form graph made of roads as edges that connect cities. The river branches joining and pouring into oceans make graphs where the join points are nodes of the graphs. Biological molecules like DNA and cell components make a graph structure whose molecules are the nodes. Similarly, social interactions like friendships and work collaborations generate social graphs, and the links among neurons in the brain make a large biological graph.

We generally characterize the network data using multi-relational graphs known as knowledge graphs. Knowledge graphs can well present the integration of data originating from different sources. An example of a knowledge graph is DBpedia [2], which is extracted from the interlinked facts in Wikipedia. It holds relations between the concepts in the encyclopedia in the shape of a graph. The Facebook social network is another knowledge graph that holds the network of friendships and people's interests. Knowledge graphs can involve a large number of entities. DRKG [3] is a biological knowledge graph that consists of approximately 5.7 million facts representing gene, drug, and disease interactions to target COVID-19.

To let machine learning algorithms intake this interconnected network of concepts as input to perform prediction/classification tasks, we need to derive representations of the semantic interactions of the graph entities in the first place. This requirement has recently led to the rise of a rapidly growing branch in the machine learning field. Graph theory calls these algorithms "representation" learning algorithms because they extract representations for graph elements such that the representations lay in the vector space. Viewing the same concept from the perspective of Algebra, we call the injective mapping of nodes and edges into vector or tensor space "embedding". We call the constructs made by such algorithms "embeddings" as well. The algorithms that encode embedding weights to represent knowledge graphs are known as knowledge graph embedding (KGE) methods or knowledge graph representation methods.

We consider both notions of graph embedding and graph representation learning in this study. We use the first term when we intend to highlight the act of mapping graph elements and the encoding process of the hidden information in knowledge graph relations. We utilize the latter in cases where we emphasize the representation of the nodes and the graph topology and when we want to highlight an actual concept that a node denotes.

KGE methods have vast applications. For example, they are commonly used in Natural Language Processing (NLP) in Relation Extraction (RE) tasks and Question Answering (QA) systems. In chemistry, they are applied in drug discovery [4], and in physics, they are used to simulate rich materials [5]. These methods are used as a recommender system as well; for example, in online retail platforms, as a product recommendation system, they suggest likely to purchase items to customers [6]. Any of these applications query the learned representations from graphs.

This study focuses on investigating and finding solutions for **challenges related to learning effective representations** for **complex knowledge graphs**. Before proceeding to the research objective of this thesis, it is essential to obtain a grasp of "complex knowledge graphs" and the "challenges related to learning effective KG representations". In the following, we clarify them in the context of a general End-to-End workflow that applies machine learning methods over knowledge graphs. Figure 1.1 illustrates the two aspects of this workflow.

The first facet of this workflow is a knowledge graph embedding method as a machine learning algorithm. The learning algorithm engages in the three steps of the workflow and is the system's primary component. The generic knowledge graph embedding workflow involves steps for Modeling and Training, Evaluation and Experiment, and Deployment. In the Modeling step of such flow, a KGE models relations in a knowledge graph with a mathematical formulation. The Training phase generates vectors or embeddings that comply with the mathematical modeling. The Evaluation and Experiment step evaluates these generated embeddings against benchmark datasets structurally similar to the primary input knowledge graph and mimics the original KG features. The Train and Evaluation steps are repeated to discover appropriate hyperparameters for the model to work optimally on the target knowledge graph. Finally, the Deploy step puts the model to work on the input knowledge graph. Generally, knowledge graph embedding methods are directly deployed in one of these directions:

- **Link Prediction**, in which it determines if there is a specific relation between two given entities.
- **Entity Classification**, in which it determines to which class an entity belongs.
- **Triple Classification**, in which it denotes the truth value of an unknown triple, i.e., to clarify whether it is a true triple.
- **Entity Resolution**, in which determines when references to real-world entities are equivalent, i.e., to resolute if they are the same.

The scope of experiments in this study is limited to evaluations in the Link Prediction category, which is the most prominent branch among the four deployment directions. Link prediction on knowledge graphs can be complex and ambiguous. The training data of KGE models are unlabeled. Therefore all the samples are positive, and they miss samples labeled as untrue (negative samples).

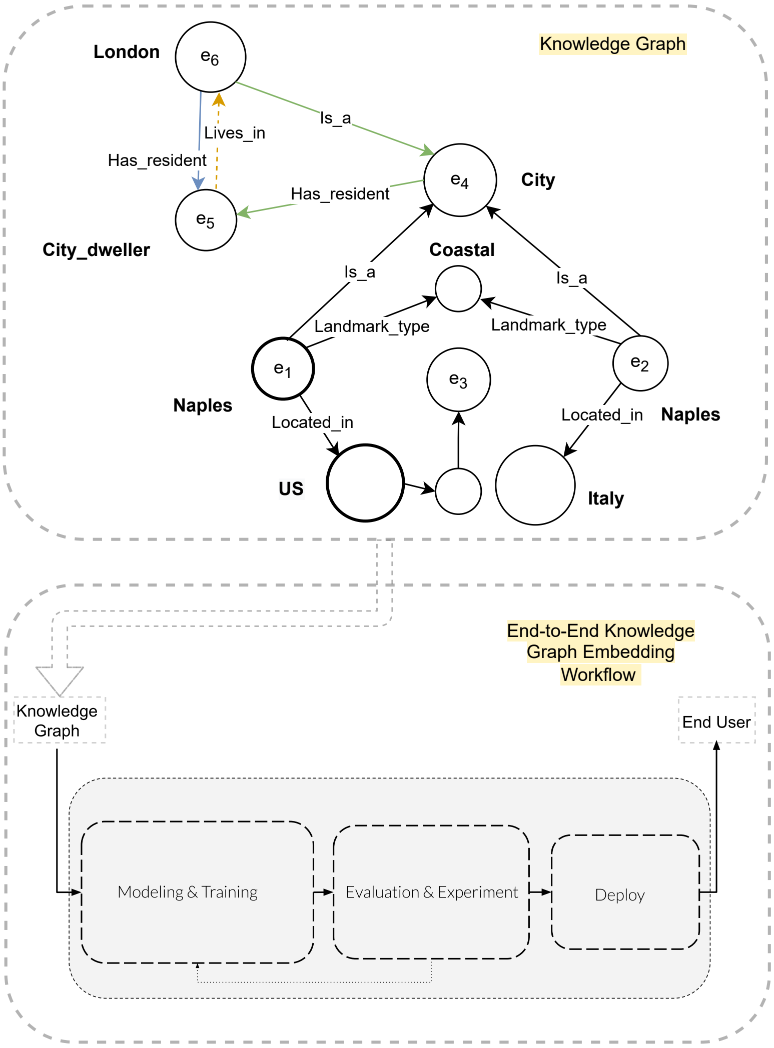


Figure 1.1: Introducing the two facets of an End-to-End machine learning workflow deploying KGE-based services.

Consequently, a positive triple gets ranked against randomly generated negative samples that are never really known to be untrue due to the open-world assumption in knowledge graphs. Furthermore, when a KGE is applied to an inductive link prediction task, the challenges of the inductive setting are added to previously mentioned complexities. In such a case, the inductive test requires link predictors to estimate relations for entities unknown to the system.

The second aspect of the workflow is the knowledge graph available for the task. Complex knowledge graphs involve various independently existing relation patterns. A relation pattern in a knowledge graph characterizes a logical connection between the semantics of comprised relations. For example, a relation r is symmetric if $\forall x, y \;\; r(x, y) \Rightarrow r(y, x)$, where x, y are entities. To understand the ability of a KGE system to approximate the triples in a knowledge graph correctly, we must consider that a KGE is bounded in its ability to learn relational patterns [7]. Therefore, the research community looks forward to more generic models covering diverse relation patterns.

Next to the intricacy of relation patterns, the knowledge graph structure and graphical specifications, the ratio of hidden relations, and the scale are among many factors that determine a model's fitness for a knowledge graph. With the introduction of very heterogeneous large-scale KGs, an embedding model can face many hidden relation patterns in one KG: Wikidata, YAGO, and DBpedia, which include data from millions of topics, present samples of complex large-scale knowledge graphs with numerous relation patterns. Therefore, it is essential to consider both the KGE element and the intake knowledge graph while designing a KGE system from a machine learning design perspective.

From the Scientific Methodology aspect, the scope of this work lies within the class of narrow Artificial Intelligence (AI) approaches, given that we base our study on statistical relational learning and optimization methods. In this line of research, to propose KGE algorithms and experiments that are data-independent, we consider a version of the algorithms and experiments that only regard data in the form of the triple-based relational facts. Nevertheless, the proposed KGE algorithms allow learning upon multi-modal data representations in application. For example, they effortlessly let the extension to encoding multi-modal graphs that involve entities designating data in textual, visual, or auditory form. Similarly, we disregard the embedding for knowledge graphs with hyper relations from the scope of this study, provided that our research is easily extendable in this direction.

1.1 Motivation

Theoretically, knowledge graphs are heterogeneous and extendable to contain facts about everything. The flexibility of knowledge graphs makes them suitable for representing complex facts and relations patterns. For instance, their network structure allows them to contain a set of relations as elaborate as a movie scenario [8] or as complicated as a mathematical formulation [9].

Predicting Links in complex knowledge graphs is a challenge for embedding models. Considering the knowledge graph at the top of Figure 1.1, suppose that the orange edge is missing from the set of available triples, and we are in the Deploy phase of the workflow for the link prediction task; we ask:

"Knowing that e_5 indicates Boris Johnson, where does he live in?"

To answer this question using a trained embedding model, we first fetch the embedding weights representing e_5 and the relation Lives_in. Then we test the existence of the triples in shape (e_5, Lives_in, ?), where we replace ? with all entities and let the embedding model estimate their score. We rank these scores, and if the embedding model has appropriately estimated the triple (e_5, Lives_in, London), it produces the best score for this triple among all triples with third entity values other than London. Assuming that the sequence of Is_a, Has_resident, Lives_in relations exist in the knowledge graph in places other than those depicted in the Figure, and the model has already faced such a pattern in the Modeling and Training phase. This sequence of relations creates a *composition* relation pattern. If the model is able to encode the composition relation pattern, it would predict other Lives_in relations whenever it faces Is_a and Has_resident in the knowledge graph.

A valuable aspect of knowledge graphs is the set of node global graphical features. The embedding of these features possibly helps to encode complex knowledge graphs better. For example, in Figure 1.1, when an embedding model attempts to generate a representation for e_1 and e_2 (Naples) based on their neighbors, it has a challenge creating distinctive embeddings because the neighbors are similar. Conversely, a model would easily separate them if it considers their different distances to the node e_3.

Current Knowledge graph embedding methods have already made significant progress in predicting hidden relations and learning the implicit relation patterns. However, these models achieve high accuracy in the link prediction task in the limited evaluations, based on a knowledge graph subsets with a limited setting. Considering the rising complexity and the extensiveness of the knowledge graphs, the objective of this work is to make highly effective algorithms for learning embeddings such that they handle the challenges of learning knowledge graphs in real-world complexity. We, therefore, need to reflect such complexity in the evaluation of our approaches as well. It is crucial that the Evaluation and Experiment phase of the workflow has already properly curated subsets of the Knowledge graph such that the model gets tested against individual relation patterns existing in the knowledge graph. Otherwise, we create a model that we are unaware of its deficiencies. It is also equally important that the evaluations on different relational modelings be performed in a stable and reproducible setup. Otherwise, the most proper model for embedding a knowledge graph would never be discovered.

Principally, a study is genuinely fruitful if it serves end-users in a real-world case. The lower right corner of Figure 1.1 states the last step of the End-to-End workflow as the Deployment and delivery. It would be beneficial to assess the application of this study in a real-world subject and review its benefits to the End User.

1.2 Problem Statement and Challenges

This Section presents the problem definition of this thesis work, followed by Challenges posed in this problem direction. To provide a clearer insight, we break down each Challenge with an example. Considering the End-to-End workflow of the machine learning applications on knowledge graphs depicted in Figure 1.1, the Research Problem of the thesis asks:

Research Problem Definition

How do we generate effective embeddings for complex knowledge graphs that effectively predict hidden links?

Based on the motivation described in Section 1.1, we identify five fundamental challenges to be tackled toward a solution for our research Problem.

Challenge 1: Learning representation of several relation patterns as the different aspects of knowledge graphs with minimal limitation on the learning capability.

Figure 1.2 displays a magnified version of the knowledge graph from Figure 1.1. In the top left corner, the orange relation Lives_in is involved in a composition relation pattern with Is_a and Has_resident. Consequently, a knowledge graph embedding model capable of predicting composition patterns can estimate the existence of this link based on the two Is_a and Has_resident relations in the Graph. The Embedding task for knowledge graphs with complicated relation patterns is challenging. Likewise, the augmentation of relation pattern types in distinct locations elevates the complexity of knowledge graphs. Furthermore, such challenges become more frequent in large-scale knowledge graphs.

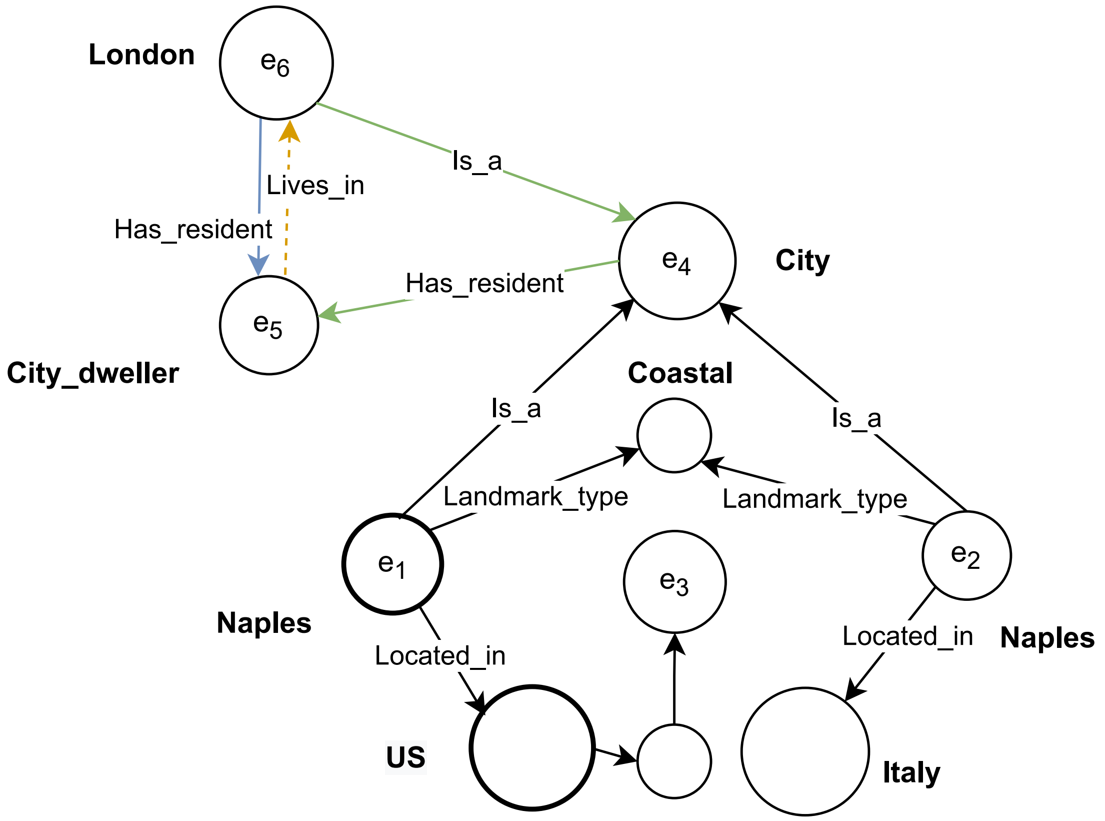


Figure 1.2: A knowledge graph consisting of information about the city Naples in Florida (US) and Naples in Italy is represented by nodes e_1 and e_2. The Challenges and Research Questions of the thesis are illustrated together in this example.

Recently a new wave of knowledge graph embedding methods has been raised that consider relation patterns in knowledge graphs [10, 11]. However, these methods are limited to only one function to implicitly learn these patterns alongside the relations. In most of these methods, the score function only allows limited encoding of a few relation patterns, hindering the other patterns. For example, DistMult [12] is an embedding model that multiplies the embedding vectors of a triple element by the element as the score function:

$$S_{DistMult} = \langle h, r, t \rangle \tag{1.1}$$

in which the relation and entities of a triple are in the form of (h, r, t). Since the multiplication of real numbers is symmetric, DistMult can not distinguish displacement of head relation and tail entities, and therefore, it can not model anti-symmetric relations. ComplEx [13] solves the issue of DistMult by replacing the tail vector in the score with its complex conjugate. With the introduction of complex-valued embeddings instead of real-valued tail to DistMult, the score of a triple in ComplEx becomes $Re(h^{\top} diag(r)\,\bar{t})$ where $\bar{t}$ means the conjugate of t and $Re(.)$ indicates the real part of a complex weight. This extension enables the ComplEx to allow non-symmetric relations; however, it becomes inefficient in encoding composition rules.

The score of TransE [14] unlike to ComplEx does not have this issue. Its score function consists of:

$$S_{TransE} = \| h + r - t \|_p \tag{1.2}$$

where $\| \, . \, \|_p$ is the p-norm. It does not limit the learning of the composition pattern; nevertheless, it forces the reflexive relations to become symmetric and transitive. Therefore, it can not effectively learn a combination of these relation patterns when one of these three patterns does not apply.

As the first Challenge to target in this study, a KGE method is required to solve these barriers, i.e., a KGE that allows training at least several crucial relation patterns. Targeting this Challenge is essential in order to embed knowledge graphs effectively. In addition to allowing the learning of variant relation patterns, such a KGE model is desired to allow the incorporation of future formulations for newly discovered relation patterns.

Challenge 2: Knowledge graph representation methods disregard the primary network features in their embedding and only consider relational learning

Besides the relational patterns that exist in knowledge graphs and are often overlooked in KGE models, an elementary yet highly neglected feature of knowledge graphs that is crucial for the KGE task is the graphical properties.

The traditional information extraction method in knowledge graphs embedding approaches local relational capturing; They learn the presence of a relationship between an entity and its hop-1 neighborhood. This learning technique ignores the notion that entities at a distance can nevertheless influence an entity's role in the whole graph. Figure 1.2 depicts an example in which the objective is to learn embeddings for e_1 and e_2 entities in the KG. The conventional relational learning approaches struggle to distinguish between the two concepts representing the cities named Naples because e_1 and e_2 have nearly identical neighbors, except that one of the cities is located in Florida (US), and the other one is not. However, we can easily differentiate e_1 and e_2 if we compare their eigenvector centrality; Because e_1 is neighbor to the US entity, with a substantial eigenvector centrality, e_1 has a greater centrality than e_2. Similarly, the shortest path of e_1 and e_2 to e_3, a member from the set of reference nodes S, is different.

Subjectively, if a KGE recognizes the centrality and distance to e_3 as meta-information, it more accurately models and ranks e_1 and e_2. Therefore, we hypothesize that considering such features enables a method to learn more than a traditional relational learning KGE approach and this extra metadata improves the ranking efficiency. Even a tiny improvement in the ability of a KGE model to distinguish entities significantly impacts its ranking efficiency over large knowledge graphs where one entity is compared to millions of other entities in this task. Thus, such improvement in a KG learning approach is crucial in real-world applications.

Because any knowledge graph is a graph in nature, and network properties are a type of information affiliated with any graph, the ubiquitous graphical features are derivable from any knowledge graph. Consequently, regarding such information besides the relational information in training not only does not limit the applicability of a learning method but also allows it to apprehend more from knowledge graphs.

As the second Challenge to consider in this study, we require generalizing KGE models to embody the graph features of KGs such as node centrality and positional indicators, e.g., the degree, Katz, or eigenvalue centrality of entities in the graph.

Furthermore, given that the downstream applications of KGE methods require trained vectors (representation weights) as the input, a graph feature-aware embedding method also is desired to produce embedding vectors, similar to the conventional embedding methods.

Challenge 3: Lack of a standard framework to generate reproducible test experiments for knowledge graph representation learning methods.

A challenge of the current benchmarks and link prediction evaluations for the KGE models is that they engage tests in different environmental settings, and comparing methods with changing parameters could favor one or another method. Therefore current studies for knowledge graph embedding have difficulty in generating reproducible experiments. This problem has been highlighted in recent link prediction tests where the small granularity of 0.001 from the unit of measurement distinguishes a better method [15]. A framework is required to generate reproductive evaluations in an unbiased environment with fixed settings. Therefore we define the lack of such a framework as the third Challenge to consider.

Challenge 4: The lack of the means to estimate the efficiency of knowledge graph embedding methods on relation patterns.

Current link prediction evaluations of KGE methods are not precisely testing the performance of a method. These evaluations rely on test datasets with a leakage problem, i.e., they include triples with a relation pattern that leaks into other relation patterns. This Challenge raises the issue of accuracy because, in a test on a relation involving two patterns, it is never clear whether the model is capable of learning both patterns or just one of them. This flaw makes it impossible to accurately test how well a model can predict a specific type of relation pattern. As a result of this gap, current benchmarks are ineffective in their goal of testing the competency of KGE methods. This Challenge is crucial in the study of knowledge graph embedding. Because to choose the best fitting KGE for different applications, it is essential to know which model best covers the relation patterns of a target knowledge graph.

The top left nodes in Figure 1.2 illustrate a case where the relation Lives_in is confirmed by both composition and inverse relation patterns. The composition relation on Is_a and Has_resident (green links) and Lives_in lead to learning of Lives_in in a model capable of learning composition patterns. However, if a model is capable of learning inverse relations, it also extracts the existence of Lives_in between e_5 and e_4 from the relation Has_residence. This case is an example of the existence of dual and more relation patterns in one link creating confusion and inaccuracy in the evaluations for the link prediction task.

This problem is omnipresent in the prominent link prediction datasets, including FB15k, WN18, WN18RR, and FB15k237. For example, WN18RR resolves the data leak of WN18 from the existing inverse relations beside other relation types. However, our observation shows that WN18RR still leaks composition pattern relations into Anti-Symmetric relations. The leakage problem of KGE evaluation datasets is the fourth Challenge we consider targeting in this study. A new group of leak-free datasets must be developed for different relation patterns. Furthermore, a new test study based on the new evaluation datasets must be conducted, and its results must be compared to the evaluation results on the current datasets. The target of these experiments must put the KGE methods under the test of the link prediction, which is one of the main evaluation types of knowledge graph embedding methods.

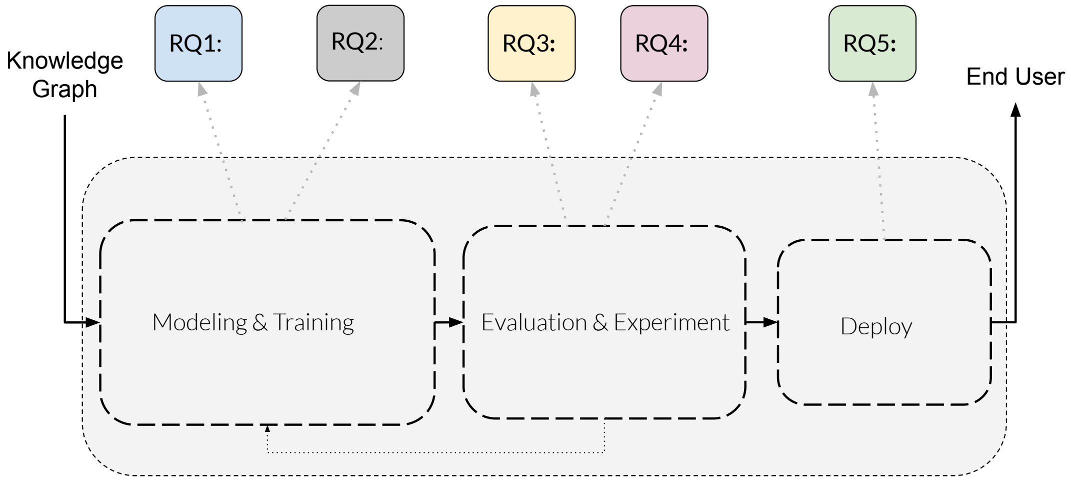


Figure 1.3: Demonstrating the connection of Research Questions to the machine learning End-to-End workflow involving link prediction on knowledge graphs.

A similar link prediction benchmark study is also missing in the inductive setting. Therefore, inductive test datasets and a set of relation patterns test experiments in the inductive setting are also required. The result of these experiments must be described via precise mathematical metrics, including AUC-PR, which is more informative than Hit reports for the inductive setting experiments.

Challenge 5: Demonstrating the capability of a KGE method that allows relation pattern encoding up to the Deploy phase of the End-to-End machine learning workflow involving a real-world input knowledge graph.

To get an insight into the usability of this study, we require to put a KGE that targets allowing relation patterns into a real-world test. The test requires the approach to be within a complete End-to-End workflow, as depicted in the Figure 1.1. As the fifth Challenge to target, this study must involve a real-world knowledge graph and show whether the relation pattern-aware embedding method is effective in such a scenario. The performance of the relation pattern aware method should be compared to a traditional KGE method as the Gold Standard method.

1.3 Research Questions

Based on the challenges, we derive the following research questions, which are addressed one by one in the succeeding chapters of the thesis. Figure 1.3 presents the connection of extracted Research Questions to the machine learning End-to-End workflow involving link prediction on knowledge graphs.

Research Question 1 (RQ1)

Does combining multiple distance-based scores targeting different relation patterns generate more effective embeddings of knowledge graphs?

With the recent decade's progress in knowledge graph embedding methods, new formulations for score functions are being developed that target a new subset of relational patterns. This research question inquires about a way to benefit various evolved models and cover the learning for all the known supported relation patterns. Simply putting the score of these models together generates conflicting formulations with limited or no learning power. We investigate how to remove this barrier practically such that we do not limit the efficiency of participating scores. A branch of KGE methods that are thoroughly studied formulate relations in knowledge graphs with geometric distances, such as TransE. This study should improve such distance-based embeddings and target modeling of different relation patterns. Such a generalization method of embedding scores should practically be evaluated to uphold this research question.

Research Question 2 (RQ2)

Does learning network features of knowledge graphs improve the efficiency of KG embedding?

In this question, we study the encoding of graphical features alongside the semantic relations inside knowledge graphs. The network features extracted from Knowledge graphs often exhibit valuable context information of entities, like their connectivity and centrality, which is commonly ignored in knowledge graph embedding methods. We investigate how a solution for the first research question would be able to learn graph features together with relations as a multi-modal learning approach. However, this information is numerical and relative to nodes in a graph. The task of learning graph features generates further challenges to the problem of multiple relation pattern learning that we target in the first research question. In contrast to the previous RQ, mathematical formulation for different graph features is a prerequisite, and a method of continuous feature modeling integration is to be found to address this research problem.

Research Question 3 (RQ3)

How can we make the experiments of embedding models reproducible?

The setup of the evaluations for KGE methods requires fair comparisons. Remarkably, the link prediction studies suffer from reproducibility problems, i.e., these methods are tested in different execution environments, e.g., with different running environments and software versions and hyper-parameters that change their outcomes. Therefore, a test generation system is required to create tests with preset hyper-parameters, test sets, and executions. The generated test experiments are required to be unalterable images that allow being restored and repeated at any time.

Research Question 4 (RQ4)

How can we accurately recognize a more effective embedding method on a specific relation pattern?

New benchmark experiments based on a new relation pattern leak-free dataset are required to address the relation pattern leakage problem in the current link prediction studies, wherein each test dataset only includes one relation pattern. The new dataset must be extracted from the existing experiment datasets to keep the evaluations based on this dataset comparable to old studies. This question also holds for inductive setting tests because, similarly, the datasets and benchmarks are missing for inducive relation pattern specific benchmarking.

Answering this question illuminates the competency question of different KGEs per relation pattern in inductive and transductive settings. In addition, it would let us understand better if a method is preferable for an application because it more accurately models a target knowledge graph given the relation patterns present in that KG.

Research Question 5 (RQ5)

Does a KGE method that allows the encoding of relation patterns tackle a real-world link prediction task more effectively than a commonly applied KGE?

The studies of KGE methods examine them over curated subsets of the existing knowledge graphs. Furthermore, they do not investigate the meaning of the link predictions. Therefore such studies inevitably do not entirely cover the challenges involved in real-world applications. RQ5 demands a real-world challenge of knowledge graph embedding approaches in the deployment phase. It specifically focuses on a KGE that allows relation patterns. It asks how such a method solves link prediction challenges more effectively than a mainstream KGE method.

A KGE method solving a link prediction challenge manifests in the real world as a recommender system on networks of data where the KGE model gives plausible suggestions to users. This study must investigate the steps required to perform this task and report the deployment workflow. The study's evaluation should deliver a qualitative analysis of the suggested links. In addition, it must provide a quantitative measure of the model's performance and compare it with a gold-standard approach.

1.4 Thesis Overview

Figure 1.4 illustrates a concise yet descriptive overview of the achieved outcomes during the conducted research. The middle section highlights the key research contributions of the thesis connected to the targeted research questions on the left side of the diagram. The right side of the diagram provides references to scientific articles addressing these contributions published throughout the whole study. In the following, we describe these contributions in detail.

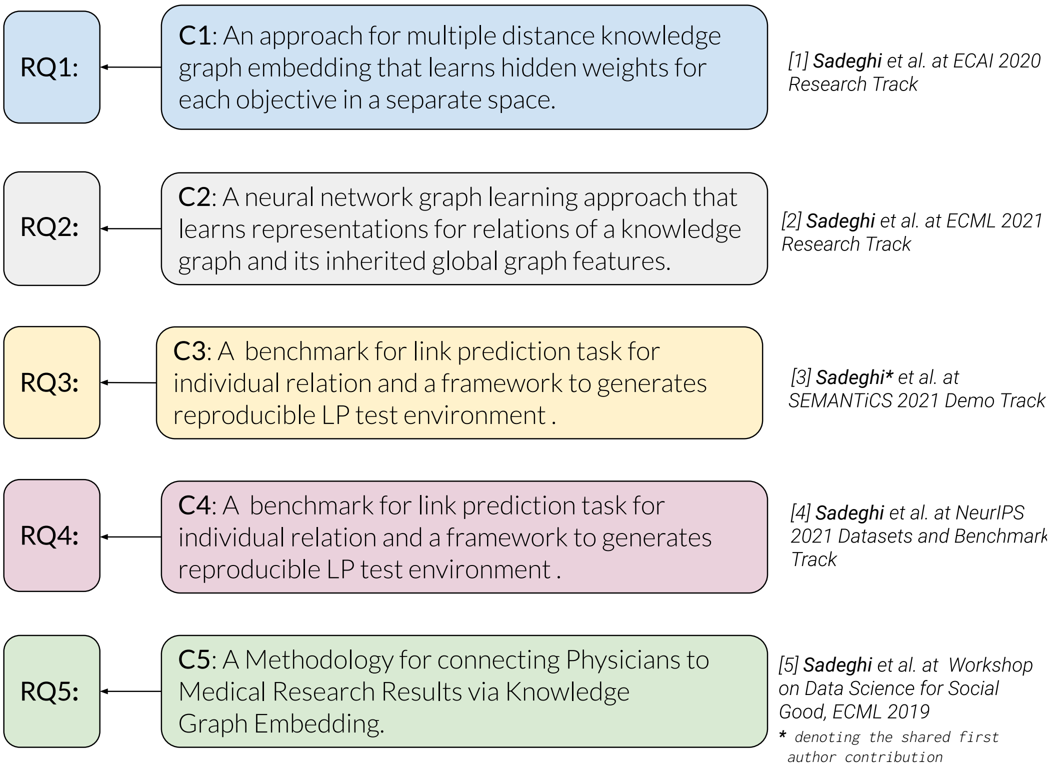

RQ1:
C1: An approach for multiple distance knowledge graph embedding that learns hidden weights for each objective in a separate space.
[1] Sadeghi et al. at ECAI 2020 Research Track
RQ2:
C2: A neural network graph learning approach that learns representations for relations of a knowledge graph and its inherited global graph features.
[2] Sadeghi et al. at ECML 2021 Research Track
RQ3:
C3: A benchmark for link prediction task for individual relation and a framework to generates reproducible LP test environment .
[3] Sadeghi* et al. at SEMANTiCS 2021 Demo Track
RQ4:
C4: A benchmark for link prediction task for individual relation and a framework to generates reproducible LP test environment .
[4] Sadeghi et al. at NeurIPS 2021 Datasets and Benchmark Track
RQ5:
C5: A Methodology for connecting Physicians to Medical Research Results via Knowledge Graph Embedding.
[5] Sadeghi et al. at Workshop on Data Science for Social Good, ECML 2019
* denoting the shared first author contribution

Figure 1.4: Contributions to the Research Questions.

1.4.1 Contributions

Contributions for RQ1

An approach for multiple distance knowledge graph embedding that targets allowing the encoding of several relation patterns in the embedding task.

To address Research Question RQ1, we suggest an approach to generalize the existing distance-based KGE methods to target different relation patterns simultaneously. We call this approach multiple distance learning, which comprises multiple desired objectives that allow learning of relation patterns individually. We formulate this novel multi-objective embedding model (MDE) by suggesting several distance-based embedding scores. We define a learning strategy to corporate these scores to benefit the effective learning of a particular pattern in each score while they do not inhibit each other. For example, one of the score sub-functions of this formulation does not produce a significant value to exemplify recognizing relations with “Composition relation pattern”. However, another sub-function is involved in the formulation specified to acknowledge such triples. Therefore, the overall output of the MDE score yet recognizes such relations. We extend this approach to several functions for different relation patterns and show that MDE allows the effective combination of these learning functions.

We put MDE under the test and show the efficiency of our method in comparison to the state-of-the-art in the link prediction task. We empirically demonstrate that the suggested approach lies within the best embedding methods. In addition, we define limit-based margin ranking loss that improves the efficiency of embedding methods by better handling the unbalanced set of positive and negative samples. We further investigate in this direction and define an algorithm to continuously find the fine limits to be used in this loss function.

Contributions for RQ2

We propose a neural network knowledge graph learning approach that learns representations for local relations of a knowledge graph and its inherited global graph features using a mathematical formulation for each learning feature.

In an attempt to revise the current KGE mechanism to address global network features without sacrificing the capacity to learn from the relational data, we propose a generalization of KGE approaches that takes advantage of the mathematical specification of graph characteristics.

We extract and normalize the network features and include these features as the numerical information about the nodes in the graph embedding. These features show the importance of nodes in graphs, such as degree centrality. We also consider the relative position of entities which indicates their closeness. This method generates a unified knowledge representation for entities in terms of embedding weights based on these properties. We evaluate this method and show how it outperforms state-of-the-art KGE models in the link prediction task. This method demonstrates competitive efficiency in the inductive link prediction setting as well. We especially highlight the significant efficiency improvement of this model over the best link prediction methods on a large-scale knowledge graph.

Contributions for RQ3

We develop a framework for the fair and reproducible link prediction evaluations of knowledge graph embedding methods.

We make a framework to generate execution environments that include the link prediction test for KGE models, focusing on generating fair experiments. These execution environments are entirely independently executable fixed Docker images. These images include a fixed execution environment, trained models with preset hyper-parameters, link prediction test modules, and fixed test sets to produce reproductive evaluation studies. This framework and its instruction are publicized on Github, and its source code and material are openly available to researchers.

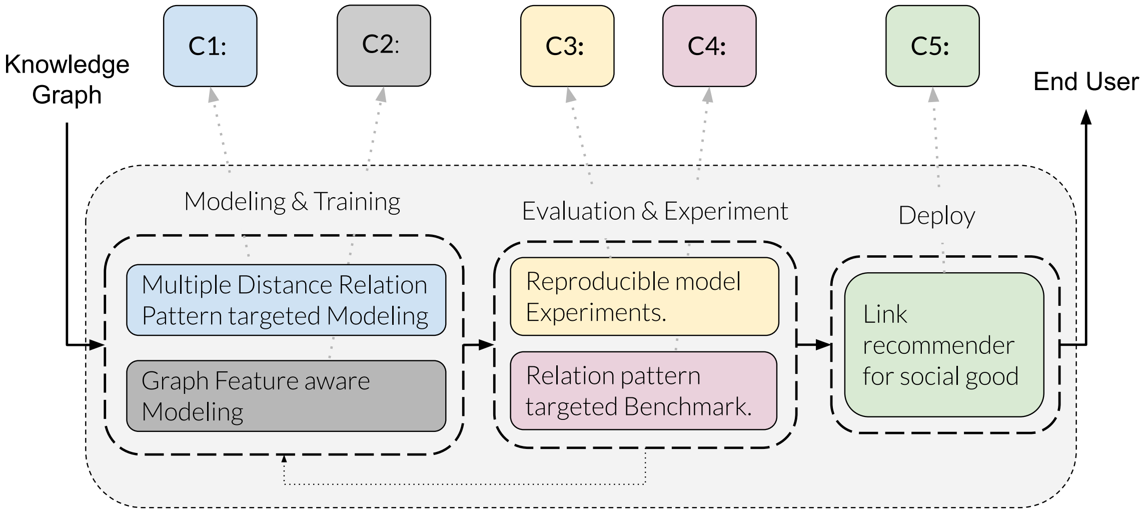

Knowledge Graph
C1:
C2:
C3:
C4:
C5:
End User
Modeling & Training
Evaluation & Experiment
Deploy
Multiple Distance Relation Pattern targeted Modeling
Graph Feature aware Modeling
Reproducible model Experiments.
Relation pattern targeted Benchmark.
Link recommender for social good

Figure 1.5: The position of the our Contributions in relation to steps in an End-to-End workflow for machine learning models involving link prediction on knowledge graphs.

Contributions for RQ4

We develop a new benchmark for the link prediction task that targets investigating KGE models on a single relation pattern basis. In addition, we apply it integrated into a framework that generates fixed test environments to perform reproducible link prediction evaluation of KGE models.

To investigate the competency and weakness of different models on the patterns precisely, we generate a set of knowledge graphs that consists of only one of the relation patterns. We study the behavior of several state-of-the-art KGE methods on each of the individual patterns. We specify this study to investigate four classes of experiments for link prediction in the degrees of inductive and transductive test entities. The four branches of generated experiment settings include fully Transductive setting, head-or-tail inductive triple setting, percentage-based semi-inductive setting, and entirely inductive setting.

The generated experiments involve 96 link prediction tests, each test including train and test and validation datasets. We particularly prune the test and validation datasets to include only one intended relation pattern, and we extract the datasets in such a way that the related training set includes triples associated with those in the test and validation datasets.

We analyze the result of state-of-the-art KGE methods on different relation patterns in various settings. In addition, we apply the generated benchmark to verify the effect of blended learning of graph features and relational learning from triples from contribution to RQ2. We demonstrate its efficiency compared to the methods that only apply traditional structural machine learning. At the center of this study, we highlight how effectively it improves the ranking performance of the MDE method that resulted from the study contributed to RQ1.

Contributions for RQ5

We develop a workflow demonstrating the MDE method's deployment to support the social good. This pipeline includes a multi-distance embedding model that performs link recommendations on the Twitter knowledge graph. This element connects physicians to the latest medical study results meaningfully relevant to them.

We create a pipeline to extract a subset of the Twitter social knowledge graph and study and demonstrate the model's usefulness in real-world applications. We showcase the model by presenting it in a scenario where it serves society and individuals regardless of their knowledge of machine learning. Our study explains all the steps of the link prediction task in this scenario, from data extraction to the prediction of links, and showcases suggesting research study results for physicians in two stages. We propose a measure to quantitatively evaluate the engaging links for physicians estimated by a KGE, and we also present an analysis that qualitatively investigates suggested links by this measure. We evaluate our proposed MDE model that allows encoding several relation patterns (from the RQ1 contribution), and we compare its effectiveness in link suggestion with TransE as a gold standard. We finally highlight the significant improvement of MDE over the gold standard model in this real-world application.

1.4.2 Publications

This thesis builds upon the following publications, which provide a basis for many of the figures, ideas, and results presented in the subsequent chapters:

- **Afshin Sadeghi**, Hirra Abdul Malik, Diego Collarana, Jens Lehmannn. *Relational Pattern Benchmarking on the Knowledge Graph Link Prediction Task*. Conference on Neural Information Processing Systems (NeurIPS) 2021. The initial study for this paper is partially covered by Hirra Abdul Malik as a part of her master thesis, where the related generated datasets did not end up in the paper due to overlooked relation pattern leaks in the code development. Besides contributing to the research investigation, Afshin Sadeghi performed the code implementation and dataset generation for the paper in a separate effort. Therefore the code, the produced datasets, and paper experiments are different from those in the master thesis. Hirra Abdul Malik assisted in the execution of the final experiments as well.

- **Afshin Sadeghi**, Diego Collarana, Damien Graux, Jens Lehmann. *Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) 2021, 548–564.

- **Afshin Sadeghi**, Xhulia Shahini, Martin Schmitz, Jens Lehmann *BenchEmbedd: A FAIR Benchmarking tool for Knowledge Graph Embeddings*. Demo track SEMANTiCS 2021. This paper is mainly collaborated by Xhulia Shahini in implementation and is partially collaborated by Martin Schmitz for the deployment setup.

- **Afshin Sadeghi**, Damien Graux, Hamed Shariat Yazdi, Jens Lehmann. *MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs*. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) 2020, 1427–1434.
- **Afshin Sadeghi**, Jens Lehmann. *Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter*. The 4th Workshop on Data Science for Social Good - SoGood 2019. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2019) 2019, 622–630.

1.5 Thesis Structure

The thesis consists of nine chapters. Chapter 1 covers the main research problem and motivates the conducted study, proposes the research questions and scientific contributions that address the research questions, and concludes with a list of published scientific papers that formally express those contributions.

Chapter 2 initially presents fundamental concepts and preliminary topics in knowledge graph embedding, relational pattern learning, and multi-objective optimization. These elements are essential for comprehending the research subject and the methodologies we have chosen to address it. Chapter 3 covers state-of-the-art community efforts in the related domains, e.g., knowledge graph embedding models, how different models deal with relational patterns, and benchmarks aimed at learning representations capable of encoding relation patterns, as well as assessment methodologies for such models.

In Chapter 4, we define a key concept for the thesis, i.e., multiple distances for knowledge graph learning, and describe a methodology for creating such models. Chapter 5 dives deeper into knowledge graph representation learning by considering the specific aspects of graphs in the multiple-distance KG embedding and shows how to effectively build a neural network learning approach on top of the multiple objectives.

Chapter 6 focuses on the relational pattern benchmarking for the link prediction task. By constructing a new set of leak-free datasets for various relation patterns, the presented study in this Chapter attempts to tackle the data leak problem of the current benchmarks and tries to provide a clearer image of the capabilities of KGE models in inductive and transductive test scenarios.

Chapter 7 describes our effort to improve the quality of KGE evaluations. Highlighting the need for fair and reproducible experiments, this Chapter demonstrates the Benchembdd framework we created that generates reproducible link prediction experiments.

In Chapter 8, using the results from the previous parts, we present and discuss a real-world application of KGE methods in Social Good using real-world data.

Finally, this thesis concludes with Chapter 9, in which we revisit the research questions and present possible future directions from two perspectives: research and applications. A summary at the start of each Chapter provides pertinent information about the topics covered.

CHAPTER 2

Preliminaries

2.1 Knowledge Representation in Networks

Knowledge graphs (KGs) encode facts about the world in a graph data structure where entities, represented as nodes, connect via relationships, which act as edges. A knowledge graph contains both schema and instance data. It provides a comprehensive representation of knowledge that spans multiple sources, domains, and levels of granularity. A knowledge graph can be open to the public, like DBpedia, or be private. An advantage of the information in the form of a knowledge graph is that it facilitates the integration of data collected from different resources.

Knowledge Graph

Knowledge graph is a network of concepts, classes, properties, relationships, and concept descriptions. It uses a relation-based knowledge representation formalism, most commonly RDF, RDF Schema, or OWL.

In contrast to tabular data, two entities can connect with different types of relations in a knowledge graph. Moreover, the network structure allows an abundant number of relations per entity. These generous benefits of KGs make them fit for a wide range of applications, such as integrating data in the industry and representing interactions in different fields, such as scientific communications.

Typically, millions of entities in a knowledge graph are linked to one another to produce a comprehensive and large-scale [16] dataset. DBpedia, YAGO, and Freebase are examples of well-known knowledge graphs. These networks represent the connection of knowledge brought together from various domains. The heterogeneous nature of knowledge graphs is highly beneficial in Machine Learning. Moreover, it allows models to work with human knowledge and language. For example, knowledge graph-based language models participate in question-answering tasks in the Natural Language Processing field.

Mathematical Formalization: Knowledge Graph

To formulate knowledge graphs mathematically, we label it as KG, comprised of two sets; The set of entities $e \in \mathcal{E}$ and a set of relations $r \in \mathcal{R}$. To comply with the open-world assumption (OWA), we assume KG as a subset of all true facts KG $\subset \xi$.

Two entities and a relation construct a triple to represent a fact in a knowledge graph. We represent a triple by (h, r, t) in which h is the head entity and t is the tail entity, and r is a relation. In Graph theory, we presume KG as a multi-relational graph. An entity in this formulation is equivalent to a node in graph theory, and an edge represents a relation. In this study, we use Node and Entity interchangeably. When we use the term “Node", we emphasize its graphical properties, and when we use the term “Entity”, we highlight the concept that entity represents.

2.2 Learning Representations for Knowledge Graphs

Graph Representation Learning methods are approaches that generate weights representing data in the form of a graph. Modeling relations in a knowledge graph produces learning weights that represent entities or predicates. Some studies name the latent representations for knowledge graphs as embeddings and the representation methods on knowledge graphs as knowledge graph embedding methods (KGE). In general, graph embedding methods studies focus on how to map the concepts in knowledge graphs into latent space effectively.

The representations produced by these methods present a knowledge graph at different levels; Often, KGE methods generate node and edge representations, while some graph learning methods synthesize weights for a sub-graph or a whole graph. In this study, our scope is limited to those methods representing facts consisting of nodes and edges. These methods always keep a mapping from an entity or a relation to their learned latent representation. The generated representation weights are primarily a (set of) vector(s), a matrix, or a tensor of numbers. In several methods, for example, in a subset of graph convolutional neural network methods, additional weights are encoded, corresponding to nodes and edges of several network layers that encode distinct neighborhood levels.

Knowledge graph Embedding, in principle, considers graphs that, in graph theory, are known as multi-relational graphs.

Mathematical Formalization: Knowledge Graph Embedding (KGE)

A relational learning model is made of an embedding function and a prediction function. The embedding function is a $f : \mathcal{E}, \mathcal{R} \longrightarrow \mathcal{Z}$, that maps entities $\mathcal{E}$ and relations $\mathcal{R}$ to d-dimensional vectors $\mathcal{Z} = \{z_1, \ldots, z_n\}$, $z_i \in \mathbb{R}$. Complementarily, a prediction function is function that given a triple $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ determines if $(\mathbf{h}, \mathbf{r}, \mathbf{t}) \in \zeta$.

We denote the embedding representation of an entity $\mathbf{h}$ with a lowercase letter h if it is a vector and with an uppercase letter H if it is a matrix.

A KG embedding model learns latent representations by forming weights directly from a score function that is modeling triples. In more complex formulations, as in graph neural networks, besides encoding weights for triples in the score function, they benefit from convolution layers that encode the combination of neighbor links into the embedding.

The encoder part of a KGE is a classifier similar to a Siamese Neural network where it learns entities based on their similarity: In each iteration, the encoder computes and compares the score of the method on a positive and a negative sample. Each sample here is a triple. When a sample is positive, the target value for the distance function of the sample embeddings is zero (or negative if using margin ranking loss). When a sample is negative, their target distance becomes one (or positive if using margin ranking loss). The training output is the embedding vectors for each KG's entities and relations. The dimension of the vector is usually much lower than the number of samples or entities. Therefore the embedding task is also known as encoding a KG into a low-dimensional space.

We define two types of KGE in this thesis:

Structure-based Embedding

A KG embedding $z_i = f : \mathcal{E}, \mathcal{R} \rightarrow \mathcal{Z}$ is attentive to network structure if it is a function of entities and relations such that it models the existence of a neighborhood of an entity e_i using relations r_i and other entities $e_j \in \mathcal{E}$.

Most knowledge graph embedding methods like QuatE [17] and RotatE [18] compute embeddings using the information describing connections between entities and, therefore, structure-based. We follow this Section with three crucial concepts in knowledge graph embedding.

Negative Sampling: Negative sampling is an essential process in KGE because the training datasets for these learning algorithms miss labeled negative samples. Knowledge graphs are based on the open-world assumption, meaning that the truth value of a statement may be true irrespective of whether or not it is known to exist in the knowledge graph.

Based on the concept of open-world assumption, a negative sample is never known to be genuinely untrue. Therefore, the datasets of the KGE task usually do not involve negative samples. Instead, the embedding methods generate negative samples while training. Usually, the training part of KGE methods generates a negative sample set based on positive samples. In this process, a KGE method usually takes a triple, then corrupts its head or tail by replacing that entity with another random entity in the datasets. Then the generated triple is checked against the positive samples and will be disregarded if it exists in the training positive sample set.

Link Prediction on Knowledge Graphs: Here we explain the details of the link prediction task and why it is a crucial part of knowledge graph embedding studies. Knowledge graphs are extensively utilized in Machine Learning, for example, to solve named entity recognition in the Natural Language Processing tasks and Question Answering. Despite all of the advances in knowledge graphs, they have reached a stalemate in terms of completeness. YAGO is missing 36.5% for gender relations mentions, and the percent of missing gender for persons has a lower coverage of 99.75%. [19].

Link Prediction is the task of estimating missing facts based on the known facts in a KG. Link Prediction based on Knowledge graph embedding targets the incompleteness of the knowledge graphs. Because a triple represents a fact, the link prediction task indicates the estimation of a missing head or a missing tail of a triple. This prediction is shown with $< h, r, ? >$ and $<?, r, t >$ and is dubbed as head and tail prediction. Any Link prediction evaluation experiment includes a negative triple generation phase. Given an incomplete triple $< h, r, ? >$, the score for all possible triples with replaced tail is inferred and compared to an entity that completes the triple from the test set. This comparison results in the score rank.

We evaluate the link prediction performance by ranking the score of each test triple against its corrupted versions where we replace the head once and once replace the tail, and then we average these scores. Based on the ranking of scores, we extract the hit at N (Hit@N), mean rank (MR), and mean reciprocal rank (MRR). In more difficult test settings, such as link prediction in the Inductive setting, we keep a fixed set of negative samples in addition to the positive sample set and compute AUC and AUC-PR over the ranking of positive and negative samples. When generating ranking tests, a false triple is only made by replacing a head or a tail, but predicates remain the same. A valid rank test includes all triples with the corrupted heads and tails by the set of all entities. We mainly report the evaluations in the filtered setting. The filter setting filters positive samples, where a generated corrupted triple is checked upon not to exist in the known test and validation and train triple sets.

2.3 Relational Pattern Learning

Relations inside a KG can form different patterns. Generally, a logic rule expresses a relation pattern. The following section describes several patterns which are frequent in knowledge graphs:

Symmetry relational pattern is a sub-category of *equivalence* pattern. Therefore, it is a binary relation that works in both directions. The relation can also be stated by the *equal to* property. For instance if a = b, then b = a.

Mathematical Definition: Symmetry

A relation r is symmetric (antisymmetric) if $\forall x, y$

$$r(x, y) \Rightarrow r(y, x) \;\; (\, r(x, y) \Rightarrow \neg r(y, x)\,).$$

A clause with such a structure has a symmetry (antisymmetry) pattern. If r^T represents the converse of r, then r is symmetric if and only if $r = r^T$. Marriage, Friendship, and Partners are a few examples of symmetric relations. The *inversion* pattern occurs when two relations are in opposite directions between two entities.

Mathematical Definition: Inverse

A relation r_1 is inverse to relation r_2 if $\forall x, y$

$$r_2(x, y) \Rightarrow r_1(y, x).$$

A clause with such a form has an inversion pattern. Parent and Child, Receive and Send, and Sell and Buy are examples of inverse relations. The *composition* pattern turns out among three relations when they form this bond through three entities:

Mathematical Definition: Composition

A relation r_1 is composed of relation r_2 and relation r_3 if $\forall x, y, z$

$$r_2(x, y) \wedge r_3(y, z) \Rightarrow r_1(x, z)$$

Relation Pattern Learning targets learning hidden relational patterns in knowledge graphs. Because different models have limited capability of relation pattern learning, based on the patterns composing a knowledge graph, one can estimate the degree of suitability of a knowledge graph to embedding models.

The relational patterns are principally not explicitly defined in a knowledge graph. Hence, to address the learning of triples with relational patterns, it is beneficial that models implicitly extract them. A lower level of support for the encoding of relation patterns is that models attempt not to limit the learning of triples with relational patterns. A knowledge graph model, in fact, not only learns representations for entities and relations but also implicitly embeds its relational patterns. Therefore, the more knowledge graph relational patterns a model supports, the better it can encode a representation of knowledge characterized by a graph.

While the large scale of knowledge graphs can be a challenge for the learning models, the complexity of relation patterns can challenge knowledge graph embedding models even on a dataset as small as a set of few triples.

2.4 Normalisation and Normal Form

The encoded metric or vector of values representing entities and relations of a graph are usually transformed into their normal form during training or before handing over to third-party applications. Normalization makes vectors trained by different models comparable, allowing other systems to recognize and use weight vectors generated by a model. Here we explain the meaning of norm for vectors and give their definition from a linear algebraic point of view. We stick to this definition throughout the thesis.

Norm is a measure indicating the size of a vector or matrix. For a vector v the length is shown with $||v||$ and for a matrix X the norm is represented by $||X||$. They are several representations of the norm for vectors and matrices. Frobenius defined a norm for matrices as the square root of $|a_{ij}|^2$ [20].

Mathematical Definition: Frobenius Norm

Frobenius matrix norm is formulated as :

$$||X||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$$

Another way to formulate this norm is the square root of the matrix trace of AA^H, where A^H is the conjugate transpose, i.e.,

$$||A||_F = \sqrt{(Tr(AA^H))}.$$

For vectors a norm is defined similarly for different root and power values.

Mathematical Definition: Vector Norm

We define vector norm $|v|_p$ for $p = 1, 2, ...$

$$|v|_p = \left(\sum_i |v_i|^p\right)^{1/p}.$$

The most common values for p are 1, 2 and 3. for p values equal to 1 and 2 the vector norm is also known as L^1-norm and L^2-norm. According to definition 5 above, L^2-norm formulation is

$$L^2 - norm = |x|_2 = \sqrt{(x_1^2 + x_2^2 + ... + x_n^2)}.$$

2.5 Optimization methods for KGE

The bottom line of the mathematical definition for any knowledge graph learning algorithm is an optimization problem for an objective function to solve. Therefore the optimization methods are the heart of any knowledge graph embedding method. The optimization algorithms find the minimum or maximum of an objective function. In the context of embedding algorithms, a better optimization produces embedding weights that comply more with the objective function and generate a smaller error. Therefore, a KGE method based on such an optimization method better separates negative and positive samples.

Between optimization methods, gradient-based approaches are the most common methods known for knowledge graph embedding. Stochastic gradient descent (SGD) is an effective gradient-based method that plays a key role in several machine learning success stories, including the Deep Learning breakthroughs and knowledge graph embedding models [21, 14, 22].

SGD iteratively optimizes approximations for the minimum or maximum of the objective function. It is a stochastic approach because instead of calculating the actual gradient gained from the entire train data set, it uses an estimate calculated from a random subset of the data. This reduction decreases the requirement of memory and computation, especially on a large dataset. We define an objective function for a knowledge graph embedding method that is discrete and can be summoned iterative as follows:

Mathematical definition: Objective Function

An objective function for knowledge graph embedding as optimization problem is in the form:

$$\boldsymbol{Q}(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w),$$

where optimization method estimates the parameter $\boldsymbol{w}$ that minimizes $\boldsymbol{Q}(w)$.

Stochastic gradient descent (SGD) steps:

We formulate SGD algorithm as:

- Choose an initial vector of parameters $\boldsymbol{w}$ and learning rate $\boldsymbol{\eta}$.
- Repeat until reaching an approximate minimum for w:
    a) Randomly shuffle samples in the training set.
    b) For $\boldsymbol{i} = 1, 2, ..., n$, do:
        $\boldsymbol{w} := w - \eta \nabla Q_i(w)$.

Adadelta algorithm [23] extended SGD by making the constant learning rate adaptive to gradient values in the previous step of the training iteration. Adam[24] in contrast, improved SGD by applying the momentum values of weights in each iteration beside the learning rate.

In this thesis, we base our approach for knowledge graph representation learning on these three optimization algorithms.

Multi Objective Optimisation: Optimization problems with more than one objective are referred to as multi-objective optimization (MOO) [25]. In contrast to single-objective optimization, a solution to a multi-objective problem may not be unique. Pareto optimality in multi-objective problems is a condition commonly applied to determine if a solution for a MOO is optimal [25].

Mathematical definition: Pareto Optimal

A point, $x^* \in X$, is Pareto optimal iff there does not exist another point, $x \in X$, such that $F(x) \leq F(x^*)$, and $F_i(x) < F_i(x^*)$ for at least one function.

A well investigated approach for multi-objective optimization is the weighted sum of objectives:

$$\boldsymbol{U} = \sum_{i=1}^{k} w_i F_i(x)$$

Where w_i is a multiplied weight and F_i is an objective. Lotfi Zadeh [26] shows that if all of the w_i weights are positive, the minimum of this sum is Pareto optimal. Therefore, minimizing U is sufficient for Pareto optimality. This formulation's Pareto Optimal condition fulfillment supports the suggested method and the contributions in Chapter 4.

2.6 Graphical Feature of Knowledge Graphs

In graph theory, an entity in a knowledge graph translates to a node in a multi-relational graph. Therefore we consider node features same as the feature of entities. A primary graphical property of a node inside a graph is its Centrality.

Centrality Value

The node's centrality value designates the node's importance concerning the whole graph. For instance, *degree* is a centrality attribute of a node that indicates the number of links incident upon it. When we consider degree as a centrality value, the higher the degree of a node is, the greater its importance in a graph.

The average length of the shortest path connecting a node to every other node in the graph is known as a node's *closeness* centrality. The *betweenness* of a node is another metric of centrality based on shortest paths. The total number of shortest paths that travel through a node is its betweenness centrality. *Katz centrality* measures the number of all nodes that can be connected through a path, while the contributions of distant nodes are penalized. It is also known as *Alpha* centrality. The *PageRank* and the *Katz* are two variants of Eigenvector centrality. Encoding graphical properties of graphs into embedding space provides a model a new vision of an embedded graph besides its structural information. Chapter 5 explores this topic in depth.

CHAPTER 3

Related Work

This Chapter reviews the state of the research and the prominent literature related to the research problem and research questions defined in Chapter 1. Based on the literature, we highlight the open gaps and the room for improvement. We first compare the current knowledge graph embedding methods with regard to their limitation and capabilities in the KGE task. We particularly pay close attention to their technique to address relation pattern learning. We then discuss and overview the previous work on improving the effectiveness of KGE methods based on the underlying knowledge graphs, including the capability in graph feature learning. In the end, we cover existing link prediction benchmarks and position our proposed solutions.

This Chapter is based on the related work sections from following publications [15, 27, 7]:

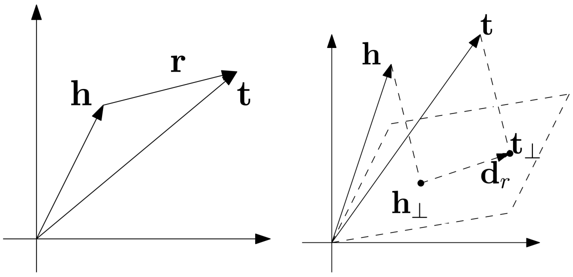
- **Afshin Sadeghi**, Damien Graux, Hamed Shariat Yazdi, Jens Lehmann. *MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs*. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) 2020, 1427–1434.

- **Afshin Sadeghi**, Diego Collarana, Damien Graux, Jens Lehmann. *Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) 2021, 548–564.

- **Afshin Sadeghi**, Hirra Abdul Malik, Diego Collarana, Jens Lehmannn. *Relational Pattern Benchmarking on the Knowledge Graph Link Prediction Task*. Conference on Neural Information Processing Systems (NeurIPS) 2021.

3.1 Knowledge Graph Embedding: Modeling Methods

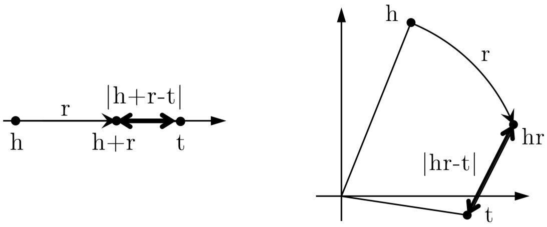
A large and growing body of literature has investigated KGE models. A typical KGE model consists of three main elements: (1) entities and relations representation in a continuous vector space, (2) a scoring function to measure KG's facts plausibility, and (3) a loss function that allows learning KGE in a supervised manner. Based on this formulation, we classify KGE models in latent distance approaches, tensor factorization and multiplicative models, and neural networks. Between the three



(a) TransE

(b) TransH

Figure 3.1: A simple illustration of TransE and TransH. The Figure is from [29].



(a) TransE models r as translation in real line.

(b) RotatE models r as rotation in complex plane.

Figure 3.2: An illustration of TransE and RotatE in complex plane. The Figure is from [30].

main elements, the score function primarily determines the capability of a KGE to encode relation patterns. The existing approaches disregard addressing a generalized learning technique for all relation patterns. Nevertheless, several models theoretically allow learning of various relation patterns. This feature empowers these methods to encode complex knowledge graphs better. We, therefore, also discuss the effect of the score functions on the capacity of KGE methods on learning relation patterns.

Latent Distance Models, e.g., the members of the Trans* [14, 28, 29] family, measure a fact's plausibility by scoring the distance between the two entities, usually after a translation carried out by the relation. TransE [14] represents the relation and entities of a triple by a vector that has this relation. TransH [29] projects the vector of head and tail to a relation-specific hyperplane. Figure illustrates TransE and TransH, where the vectors representing the head and tail are projected to a relation-specific hyperplane. This extension allows the model to learn a weight W_r per relation to better specify the connection of a head to a tail using a relation in a positive training sample.

Similarly, TransR [28] follows the idea of relation-specific spaces and extends the distance function. RotatE [30] combines translation and rotation. RotatE models relations as rotations from head to tail entities in the complex space and uses the Hadamard product in the score function to do these rotations. Figure compares TransE and RotatE in a complex plane. This extension allows the model to encode an extended set of interactions between the relations in different relation patterns.

A direct interpretation of the definition for distance-based methods is that the embedding representations for closely related entities lay close to each other. This feature makes this class of embeddings fit for entity matching applications. Designing an effective score function for such models is the subject of many recent studies in knowledge graph-based entity matching [31, 32, 33, 34].

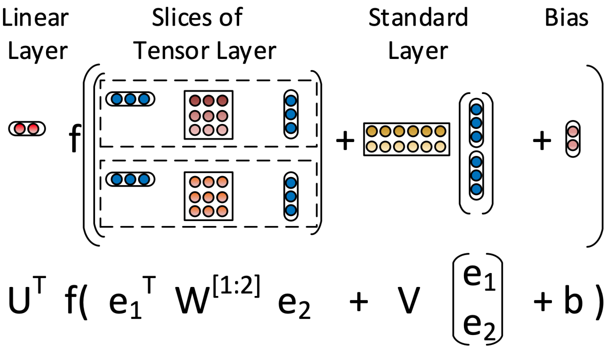


Figure 3.3: A visualization of the neural tensor network. Each dashed box represents an slice of the tensor, in this case there are k = 2 slices. The Figure is from [37].

Tensor Factorization and Multiplicative Approaches define the score of triples via pairwise multiplication of embeddings. RESCAL [35] is a early multiplicative model that proposes loss-based tensor factorization to approximate an adjacency tensor by vector-matrix product of $\mathbf{h}^{t}$ **r t**. DistMult [12] multiplies the embedding vectors of a triple element by element (h, r, t) as the objective function. However, DistMult fails to distinguish displacement of head relation and tail entities, and therefore, it cannot model anti-symmetric relations. Finally, ComplEx [13] is an extension of DistMult that performs the multiplication in the complex space. This solution allows learning of anti-symmetric relations, which is the DistMult's issue; however, it inhibits learning the composition relation pattern that DistMult permits in the first place.

Neural Network Methods train a neural network to learn the interaction of the **h**, **r** and **t**. Unlike previous methods, connecting artificial neurons in different layers allows Graph Neural Network (GNN) aggregate node formation using a message-passing architecture.

ER-MLP [36] is a two-layer feedforward neural network considering h, r and t vectors in the input. NTN [37] is a neural tensor network meaning that it replaces a standard linear layer from the conventional neural networks with a bilinear tensor layer. In the tensor layer, NTN concatenates head h and tail t vectors and feeds them to the first layer that has r as weight. In another layer, it combines h and t with a tensor R that represents **r** and finally, for each relation, it defines an output layer r to represent the relation embeddings. Figure 3.3 shows a visualization of the neural tensor network.

In SME [38] relation r is once combined with the head h to get $g_u(h, r)$, and similarly it is combined with the tail t to get $g_v(t, r)$. SME defines a score function by the dot product of this two functions in the hidden layer. In the linear SME, $g(e, r)$ is equal to $M_u^1 e + M_u^2 r + b_u$, and in the bilinear version, it is $M_u^1 e \circ M_u^2 r + b_u$. Here, M refers to weight matrix and b is a bias vector.

Graph convolutional networks (GCNs) are a subset of convolutional neural networks (CNNs) that can handle non-Euclidean graph data. Most GCN methods follow the message passing mechanism from MPNN [39]. Message passing is an aggregate function of embedding weights and allows nodes to equally influence their hop-one neighbors. Mostly GCNs are used for node classification. However, recent extensions of GCNs allow the encoding of knowledge graphs [40, 41, 42, 43].

Hybrid Methods have recently raised, such as CompGCN [44] which is a convolutional graph network benefiting from score functions proposed in embedding methods. MDE$_{nn}$ [15] and GFA-NN [27] are likewise two models in this branch which we propose and discuss in detail in the Chapters 4 and 5. The hybrid methods benefit from the most effective score functions from knowledge graph embedding methods and the neural network structure.

3.2 Advances in Knowledge Graph Elements

The last Section reviewed different types of KGE methods and discussed how the formulation of KGE score functions enables or inhibits the learning of different relation patterns. This factor significantly influences the effectiveness of KGE methods. Here we review related works categorized by three other principal elements of knowledge graph embedding that recently evolved to improve the performance of the KGE methods.

Negative Sampling: Empirical studies show that baseline methods like TransE and DistMult are more effective with proper sampling than many more recent KGE methods [18, 45]. While earlier KGE methods compare only one negative sample per positive sample in calculating their score functions, novel methods improved the embedding efficiency by extending the negative sampling technique. A technique to improve negative sampling is altering the probabilistic function for making negative samples. While most methods uniformly select replacement entities to make negative samples, TransH [29] applies Bernoulli sampling in replacing the head or tail with different probabilities for one-to-many, many-to-one, and many-to-many relations. This extension consequently reduces falsely labeled negative samples by allowing the extraction of negative samples per positive sample and weighting them based on the frequency of links.

KBGAN [46] proposes a Generative Adversarial Network (GAN) schema for generating negative samples based on triple weights extracted using the Bernoulli sampling. This sampling approach applies a twist to the margin ranking loss that allows a balanced combination of loss values for negative and positive samples by assigning them probabilistic weights using the Softmax function. RotatE simplifies this negative sampling technique by integrating the generator component inside the KGE method instead of executing a discrete generator in parallel. This extension is more computationally efficient and allows the calculation of numerous negative samples per positive sample. This technique is particularly effective on KGE methods whose score functions are defined based on margin ranking loss, such as DistMult and TransE. Our suggested KGE methods in Chapters 4 and 5 involve one-to-one negative sampling with limit-based loss, in addition to adversarial negative sampling coupled with margin ranking loss.

In comparison, Neural network designs for KGE [44, 13, 47] conventionally use Mean squared error (MSE) as the loss function that allows comparing a batch of positive samples against all possible negative samples in one iteration. However, this method would be more efficient if their scores for triples get probabilistic weight by a Softmax function [46].

Graph Feature Learning: Several studies have investigated the benefits of using graph features to bridge the graph structure gap and the numeric vector space. Muzzamil et al. [48] defined a Fuzzy Multilevel Graph Embedding (FMGE), an embedding of attributed graphs with many numeric values.

P-GNN [49] incorporates positional information by sampling anchor nodes and calculating their distance to a given node. Finally, it learns a non-linear distance weighted aggregation scheme over the anchor nodes. In Chapter 5, we propose an model that generates embeddings aware of graph features of an underlying Knowledge graph. In Section 5.3.1 we present an in-depth comparison of P-GNN with GFA-NN. Traditionally, GNN approaches only encode nodes' local features (similar to the modeling schema of KGEs) while focusing on neighbor nodes; P-GNN extends them to include the distances to anchor nodes. However, our approach also learns nodes' features regarding the whole graph, known as global node properties.

Inductive Support: Link Predictors in the Inductive task estimate relations for entities not declared in the training data. KGE Methods specified for this type of link prediction, try to learn more context information from the knowledge graph related to the asked entity. Our GFA-NN [27] method presented in Chapter 5 learns several entities centrality values and positional features for entities which are contextual information related to entities. Similarly, CatE [50] encodes the degree and positional features to target Inductive test triples. However, this model uses the score function of TransE to model triples, while GFA-NN is based on multiple objective optimizations.

To extract context information of nodes, GraIL [51] encodes the sub-grubs involving targeted entities and ensembles the sub-graph embeddings with the embeddings produced by the standard KGE methods, for example, TransE, to produce a better result in the link prediction tests. BERTRL [52] borrows external contextual information related to entities from BERT language model [53]. However, this model is limited to cases where textual information related to entities exists. DKRL[54] similarly learns external textual external information related to entities to predict links in the Inductive setting, and in addition, it embeds triples beside embedding entity and predicates. Using external context information is not limited to associated textual information of entities. For example, in IKRL [55], researchers involve image embedding to provide external information in the process of knowledge graph embedding.

INDIGO [56] uses the strategy of context information learning for the task of knowledge graph completion. In the knowledge graph completion task, the KGE predicts missing triples for an incomplete knowledge graph. This method has an encoder/decoder structure, where it applies a neural network on the embedding weights after encoding a knowledge graphs. Finally, researchers that propose LAN [57] for the knowledge graph completion task suggest considering the contextual information of redundant neighbors and the logic relation patterns for Inductive knowledge graph embedding.

3.3 Link Prediction Benchmarking Datasets

Benchmark datasets are a means to compare and evaluate the link prediction models, parameters, and procedures. The benchmarked dataset, as described by [58] is helpful for two fundamental analyses: efficiency and effectiveness. We adopt and extend their analysis types in this study. Our efficiency analysis includes assessing the time consumption for training and prediction. Meanwhile, our analysis investigates the effectiveness of KGE models in 3 branches. The first is the number of negative samples tested against a positive sample. The second branch is the support for graph features learning, for example, path learning. The third effectiveness test branch considers the performance in relational pattern learning.

The datasets for link prediction evaluation are often extracted by subsampling real-world KGs. This data then becomes split into training, validation, and test sets. We list the four most well-known link prediction datasets related to our work:

- Bordes et al. [14] extracted FB15k by filtering all of the FreeBase[1] entities featuring more than 100 times and excluded literals. FreeBase is a large-scale knowledge graph with approximately 44 million entities and 1.9 billion triples, where the entities refer to topics in common human knowledge such as actors and cities.

- Bordes et al. [14] in the same study extracted WN18 from WordNet 4. The entities in WordNet represent word senses, and the relations indicate the lexical relationships between the synsets.

- FB15k-237 [59] is a subset of FB15k that excludes the triples that leak inverse relation patterns between the train and test sets. Nevertheless, this dataset has leakage in the other relation patterns.

- Similar to generation of FB15k-237, Dettmers et al. [47] made WN18RR by filtering out the inverse relations from WN18. This dataset likewise suffers from leakage between other relation patterns.

Open graph benchmark [60] is an extensive study that generates several datasets for different KGE-based tasks, including link prediction. We use the biological ogb-biokg dataset in our experiments of Chapter 5 to test how well the proposed method in this Chapter predicts links on a large-scale dataset, considering that the training data of ogb-biokg includes approximately 4.7 million triples.

In the direction of link prediction evaluation in the Indicative setting, the GraIL [51] study extracts subsets of WN18RR, FB15k-237, and NELL-995 datasets. We include several of these datasets in the experiments section of Chapter 5 to evaluate if embedding graphical features of a knowledge graph supports the model in the Inductive setting. For testing KGE methods specifically in the n-ary link prediction setting, the TRFR [61] study extracts NELL-995-3 from NELL-995 [62] and Similarly the NaLP [63] study extracts WikiPeople from the Wikidata[2] dataset. In these benchmarks, the percentage of n-ary relations is specified by filtering other triples. Therefore, they include a higher percentage of n-ary relations compared to the regular link prediction datasets.

In Chapter 7 we generate a collection of datasets and experiments to fill the gap of lacking leak-free link prediction benchmarks and test the capability of KGE methods in learning relation patterns. In this study, we keep the study of ogb [60] and NELL-995 [62] datasets and the studies for n-ary link prediction setting as our ultimate guidance and standard to support our benchmarking task. There we cover the experiments in both Inductive and Transductive settings. The study of n-ary link prediction datasets inspired us to include percentage-based datasets as a part of our benchmark.

After reviewing the preliminaries and related work in knowledge graph embedding to this point, we start the core part of this study in the following by proposing multiple distance embeddings. This approach targets the crucial Challenge 1 posed in the Introduction by allowing the encoding of relation patterns for complex knowledge graphs.

[1] https://developers.google.com/freebase/

[2] https://www.wikidata.org/wiki/Wikidata:Main_Page

CHAPTER 4

Multiple Distance Embeddings for Link Prediction in Knowledge Graphs

Over the past decade, knowledge graphs have become popular for capturing structured domain knowledge. Relational learning models enable the prediction of missing links inside knowledge graphs. More specifically, latent distance approaches model the relationships among entities via a distance between latent representations. Translating embedding models (e.g., TransE) are among the most popular latent distance approaches which use one distance function to learn multiple relation patterns. However, they are mostly ineffective in capturing symmetric relations since the representation vector norm for all the symmetric relations becomes equal to zero. They also lose information when learning relations with reflexive patterns since they become symmetric and transitive.

In this Chapter, we propose the Multiple Distance Embedding model (MDE) that addresses these limitations and a framework to collaboratively combine variant latent distance-based terms. Our solution is based on two principles: 1) we use a limit-based loss instead of a margin ranking loss, and 2) by learning independent embedding vectors for each of the terms, we can collectively train and predict using contradicting distance terms. We further demonstrate that MDE allows modeling relations with (anti)symmetry, inversion, and composition patterns. We propose MDE as a neural network model that allows us to map non-linear relations between the embedding vectors and the expected output of the score function. Our empirical results show that MDE performs competitively to state-of-the-art embedding models on several benchmark datasets. We specifically target this research question in this Chapter:

Research Question 1

Does combining multiple distance-based scores targeting different relation patterns generate more effective embeddings of knowledge graphs?

Overall, the contributions in this Chapter are as follows:

- Proving the theoretical analysis of current knowledge graph embedding models and pointing out their limitation on learning different relation patterns.

- Developing MDE knowledge graph embedding method and showing that MDE allows encoding several relational patterns.
- Showing the proposed method is extensible to other relation patterns if the extension is properly formulated.
- Suggesting limit-based loss function for knowledge graph embedding.
- Proposing an algorithm to actively search the limits for the limit-based loss function to use in embedding models.
- Empirical evaluating our method, where it performs competitively to the state-of-the-art in link prediction experiments.

This Chapter is based on the following publication [15]:

- **Afshin Sadeghi**, Damien Graux, Hamed Shariat Yazdi, Jens Lehmann. *MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs*. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) 2020, 1427–1434.

While machine learning methods conventionally model functions given sample inputs and outputs, a subset of Statistical Relational Learning (SRL) [64, 65] approaches specifically aim to model “things” (entities) and relations between them. These methods usually model human knowledge, which is structured in the form of multi-relational knowledge graphs (KG). Knowledge graphs allow semantically rich queries and are used in search engines, natural language processing (NLP), and dialog systems. However, they usually miss many of the true relations [66]; therefore, predicting missing links/relations in KGs is a crucial challenge for SRL approaches.

Practically, a KG usually consists of a set of facts. We assume any fact is representable by a triple (head, relation, tail) where heads and tails are called entities. Distance-based KG embeddings are popular among the SRL models because of their simplicity, the low number of parameters, and efficiency on large-scale datasets. Specifically, their simplicity allows their integration into many models. Previous studies have integrated them with logical rule embeddings [67], have adopted them to encode temporal information [68] and have applied them to find equivalent entities between multi-language datasets [69].

Soon after the introduction of the first multi-relational distance-based method, TransE [14], it was acknowledged that it is ineffective in learning symmetric relations since the norm of the representation vector for all the symmetric relations in the KG becomes close to zero. This shortcoming limits the model from distinguishing different symmetric relations in a KG well. To extend this model, many variations were studied afterward, e.g., TransH [29], TransR [28], TransD [70], and STransE [71]. Even though they solved the issue of symmetric relations, they introduced another limitation: these models were no longer effective in learning the inversion and composition relation patterns that originally TransE could handle.

Besides, as noted in [72, 30], within the family of distance-based embeddings, reflexive relations are usually forced to become symmetric and transitive. In this study, we take advantage of independent vector representations of vectors that enable us to view the same relations from different aspects and put forward a translation-based model that addresses these limitations and allows the learning of all three relation patterns. In addition, we address the issue of the limit-based loss function in finding an optimal limit and suggest an updating limit loss function to be used complementarily to the current limit-based loss function, which has fixed limits. Moreover, we frame our model into a neural network structure that allows it to learn non-linear patterns for the limits in the limit-based loss, improving the generalization power of the model in link prediction tasks.

The model performs well in the empirical evaluations, competing against state-of-the-art models in link prediction benchmarks. In particular, it outperforms state-of-the-art models on Countries [73] benchmark, which is designed to evaluate composition pattern inference and modeling.

Since our approach involves several elements that model the relations between entities as the geometric distance of vectors from different views, we dubbed it **m**ultiple-**d**istance **e**mbeddings (MDE)[1].

The rest of this Chapter is structured as follows: In Section 4.1, we summarize the related efforts in KGE with an outlook on their effectiveness in allowing learning of relation patterns. Then we present the MDE model in Section 4.2 and describe the extensions of the model, including a hyperparameter search algorithm for the loss function and a Neural Network framing of MDE in Section 4.3. We report on the experiments in Section 4.4 before concluding.

4.1 Relation Pattern Coverage in KGE models

In this Section, we review how previous KGE models allow at least the learning of one or two relation patterns in their embedding; however, they mostly block the learning of other patterns.

Tensor Factorization and Multiplicative Models define the score of triples via pairwise multiplication of embeddings. DistMult [12] simply multiplies the embedding vectors of a triple element-by-element $\langle h, r, t\rangle$ as the score function. Since the multiplication of real numbers is symmetric, DistMult can not distinguish displacement of head relation and tail entities, and therefore, it can not model anti-symmetric relations.

ComplEx [13] solves the issue of DistMult by the idea that the complex conjugate of the tail makes it non-symmetric. By introducing complex-valued embeddings instead of real-valued embeddings to DistMult, the score of a triple in ComplEx is $Re(h^\top diag(r)\bar{t})$ with $\bar{t}$ the conjugate of t and $Re(.)$ is the real part of a complex value. ComplEx is not effective in encoding composition rules [30]. In RESCAL [22] instead of a vector, a matrix represents the relation r, and performs outer products of h and t vectors to this matrix so that its score function becomes $h^\top Rt$. A simplified version of RESCAL is HolE [74] that defines a vector for r and performs circular correlation of h and t has been found equivalent [75] to ComplEx.

Another tensor factorization model is Canonical Polyadic (CP) [76]. In CP decomposition, each entity e is represented by two vectors $h_e, t_e \in \mathbb{R}^d$, and each relation r has a single embedding vector $v_r \in \mathbb{R}^d$. MDE is similarly based on the idea of independent vector embeddings. A study [77] suggests that in CP, the independence of vectors causes the poor performance of CP in KG completion, however, we show that the independent vectors can strengthen a model if they are combined complementarily.

SimplE [72] analogous to CP, trains on two sets of subject and object entity vectors. SimplE's score function, $\frac{1}{2}\langle h_{e_i}, r, t_{e_j}\rangle + \frac{1}{2}\langle h_{e_j}, r^{-1}, t_{e_j}\rangle$, is the average of two terms. The first term is similar to DistMult. However, its combination with the second term and using the second set of entity vectors allows SimplE to avoid the symmetric issue of DistMult. SimplE allows learning of symmetry,

[1] The complete code and the experimental datasets are available from: https://github.com/mlwin-de/MDE

anti-symmetry, and inversion patterns. However, it is unable to effectively encode composition rules, since it does not model a bijection mapping from h to t through relation r.

In **Latent Distance Approaches** the score function is the distance between embedding vectors of entities and relations. In the view of social network analysis, [78] originally proposed distance of entities $-d(h, t)$ as the score function for modeling uni-relational graphs where $d(., .)$ means any arbitrary distance, such as Euclidean distance. SE [79] generalizes the distance for multi-relational data by incorporating a pair of relation matrices into it. TransE [14] represents relation and entities of a triple by a vector that has this relation

$$S_1 = \| h + r - t \|_p \tag{4.1}$$

where $\| \, . \, \|_p$ is the p-norm. To better distinguish entities with complex relations, TransH [29] projects the vector of head and tail to a relation-specific hyperplane.

Similarly, TransR follows the idea with relation-specific spaces and extends the distance function to $\| M_r h + r - M_r t \|_p$. In short, the members of the Trans* [14, 28, 29] family measure a fact's plausibility by scoring the distance between the two entities, usually after a translation carried out by the relation.

RotatE [30] combines translation and rotation and defines the distance of a t from tail h, which is rotated the amount r as the score function of a triple $-d(h \circ r, t)$ where $\circ$ is Hadamard product. This method allows for modeling relations with (anti)symmetry, inversion, and composition patterns; however, do not cover learning the Transitivity relation pattern.

A new distance-based KGE method [80] allows learning of the Transitivity relation pattern. Despite its mediocre efficiency, this effort is notable because it covers a relation pattern that its learning was inhibited in the previously reviewed KGE methods.

Unlike the earlier mentioned methods, the **Neural Network-based methods** learn KGE by connecting artificial neurons in different layers. Graph Neural Network (GNN) aggregate node formation using a message-passing architecture. Neural networks are black-box models, i.e., the structure of the function being approximated by them is not clear. Therefore, in theory, their learned weights are not directly interpretative to relation patterns. However, empirical evaluation methods specially made for relation pattern learning can mitigate this gap. We will cover a study to generate such evaluations in Chapter 7. There we also include the pattern relation learning analysis of state-of-the-art neural networks for knowledge graph learning.

In summary, the reported works only focus on modeling the relations of the knowledge graphs, and allowing relation patterns was a side benefit for them. However, our study attempts to directly address allowing learning of different relation patterns by a specific structure design and modeling functions tailored to the task.

4.2 MDE: Multiple Distance Embeddings

The score function of MDE involves multiple terms. We first explain the intuition behind each term and then explicate a framework that we suggest to effectively utilize them such that we benefit from their strengths and avoid their weaknesses.

Inverse Relation Learning: Inverse relations can be a strong indicator in knowledge graphs. For example, if $IsParentOf(m, c)$ represents that a person m is a parent of another person c, then this could imply $IsChildOf(c, m)$ assuming that this represents the person c being the child of m.

This indication is also valid in cases when this only holds in one direction, e.g. for the relations $IsMotherOf$ and $IsChildOf$. In such a case, even though the actual inverse $IsParentOf$ may not even exist in the KG, we can still benefit from inverse relation learning. To learn the inverse of the relations, we define a score function S_2 :

$$S_2 = \| t + r - h \|_p \tag{4.2}$$

Symmetric Relations Learning: It is possible to easily check that the formulation $\| h + r - t \|$ allows[2] learning of anti-symmetric pattern but when learning symmetric relations, $\| r \|$ tends toward zero which limits the ability of the model in separating entities especially if symmetric relations are frequent in the KG. For learning symmetric relations, we suggest the term S_3 as a score function. It learns such relations more effectively despite it is limited in the learning of anti-symmetric relations.

$$S_3 = \| h + t - r \|_p \tag{4.3}$$

Lemma 1. S_1 allows modeling anti-symmetry, inversion, and composition patterns, and S_2 allows modeling symmetry patterns.

Proof. Let r_1, r_2, r_3 be relation vector representations and e_i, e_j, e_k are entity representations. A relation r_1 between (e_i, e_k) exists when a triple (e_i, r_1, e_k) exists and we show it by $r_1(e_i, e_k)$. Formally, we have the following results:

Anti-symmetric Pattern. If $r_1(e_i, e_j)$ and $r_1(e_j, e_i)$ hold, in equation 4.1 for S_1, then:

$$e_i + r_1 = e_j \quad \wedge \quad e_j + r_1 \neq e_i \quad \Rightarrow \quad e_i + 2r_1 \neq e_i$$

Thus S_1 allows encoding of relations with anti-symmetric patterns.

Symmetric Pattern. If $r_1(e_i, e_j)$ and $r_1(e_j, e_i)$ hold, for S_3 we have:

$$e_i + e_j - r_1 = 0 \ \wedge \ e_j + e_i - r_1 = 0 \ \Rightarrow \ e_j + e_i = r_1$$

Therefore S_3 allows encoding relations with symmetric patterns. For S_1 we have:

Inversion Pattern. If $r_1(e_i, e_j)$ and $r_2(e_j, e_i)$ hold, from Equation 4.1 we have:

$$e_i + r_1 = e_j \quad \wedge \quad e_j + r_2 = e_i \quad \Rightarrow \quad r_1 = -r_2$$

Therefore S_1 allows encoding relations with inversion patterns.

Composition Pattern. If $r_1(e_i, e_k)$, $r_2(e_i, e_j)$ and, $r_3(e_j, e_k)$ hold, from equation 4.1 we have:

$$e_i + r_1 = e_k \ \wedge \ e_i + r_2 = e_j \ \wedge \ e_j + r_3 = e_k \ \Rightarrow \ r_2 + r_3 = r_1$$

Thus S_1 allows encoding relations with composition patterns. □

[2] We used the term “it allows” to imply that the encoding of such patterns does not inhibit the learning of relations having a particular pattern. Meanwhile, in the literature, SimplE uses “it can encode” and RotatE uses “the model infers”.

Relieving Limitations on Learning of Reflexive Relations:
A previous study [72] highlighted the common limitations of TransE, FTransE, STransE, TransH, and TransR for learning reflexive relations where these translation-based models force the reflexive relations to become symmetric and transitive.

To relieve these limitations, we define S_4 as a score function that is similar to the score of RotatE i.e., $\| h \circ r - t \|_p$ but with the Hadamard operation on the tail. In contrast, to RotatE which represents entities as complex vectors, S_4 only holds in the real space:

$$S_4 = \| h - r \circ t \|_p \tag{4.4}$$

Lemma 2. The following restrictions of translation-based embeddings approaches do not apply to the S_4 score function. R1: if a relation r is reflexive, on $\Delta \in \mathcal{E}$, r it will be also symmetric on Δ. R2: if r is reflexive on $\Delta \in \mathcal{E}$, r it will be also be transitive on Δ.

Proof. R1: For such reflexive r_1, if $r_1(e_i, e_i)$ then $r_l(e_j, e_j)$. In this equation we have:

$$e_i = r_1 e_i \wedge e_j = r_1 e_j \Rightarrow r_1 = U \nRightarrow e_i = r_1 e_j$$

where U is unit tensor.

R2: For such reflexive r_1, if $r_1(e_i, e_j)$ and $r_l(e_j, e_k)$ then $r_1(e_j, e_i)$ and $r_l(e_k, e_j)$. In the above equation we have:

$$\begin{aligned} e_i = r_1 e_j \wedge e_j = r_1 e_k &\Rightarrow e_i = r_1 r_1 e_j e_k \wedge r_i = U \\ &\Rightarrow e_i = e_j e_k \\ &\nRightarrow e_i + e_k = r_l \end{aligned}$$

□

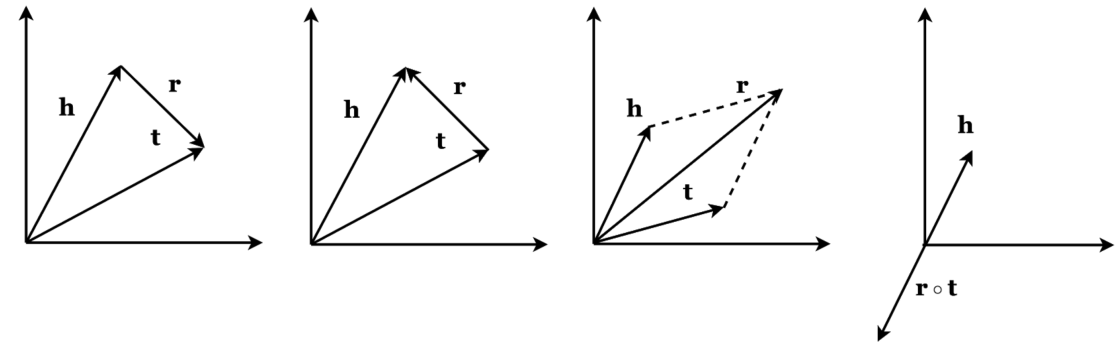
Model Definition: To incorporate different views to the relations between entities, we define these settings for the model:

- Using limit-based loss instead of margin ranking loss.
- Each aggregated term in the score represents a different view of entities and relations with an independent set of embedding vectors.
- In contrast to ensemble approaches that incorporate models by training independently and testing them together, MDE is based on multi-objective optimization [81] that jointly minimizes the objective functions.

However, when aggregating different terms in the score function, the summation of opposite vectors can cause the norm of these vectors to diminish during the optimization. For example, if S_1 and S_3 are added together, the minimization would lead to relation(r) vectors with zero norm value. To address this issue, we represent the same entities with independent variables in different distance functions.

Based on CP, MDE considers four vectors $e_i, e_j, e_k, e_l, \in \mathbb{R}^d$ as the embedding vector of each entity $\mathbf{e}$, and four vectors $r_i, r_j, r_k, r_l \in \mathbb{R}^d$ for each relation $\mathbf{r}$.

The score function of MDE for a triple ($\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$) is defined as weighted sum of listed score functions:



$\| h_i + r_i - t_i \|_p$ $\| t_k + r_k - h_k \|_p$ $\| h_j + t_j - r_j \|_p$ $\| h - r \circ t \|_p$

Figure 4.1: Geometric illustration of the translation terms considered in MDE.

$$f_{MDE} = w_1 S_1^i \; + \; w_2 S_2^j \; + \; w_3 S_3^k \; + \; w_4 S_4^l \; - \psi \tag{4.5}$$

where $\psi, w_1, w_2, w_3, w_4 \in \mathbb{R}$ are constant values. Figure 4.1 displays the geometric illustration of the four translation terms considered in MDE.

In the following, we show using ψ and limit-based loss, the combination of the terms in Equation (4.5) is effective, such that if one of the terms recognizes if a sample is true F_{MDE} would also recognize it.

Limit-based Loss: Because margin ranking loss minimizes the sum of error from directly comparing the score of negative to positive samples, when applying it to translation embeddings, it is possible that the score of a correct triplet is not small enough to hold the relation of the score function [82]. To enforce the scores of positive triples become lower than those of negative ones, [82] defines limited-based loss which minimizes the objective function such that the score for all the positive samples becomes less than a fixed limit. [33] extends the limit-based loss so that the score of the negative samples becomes greater than a fixed limit. We train our model with the same loss function which is:

$$loss = \beta_1 \sum_{\tau \in \mathbb{T}^+} [f(\tau) - \gamma_1]_+ + \beta_2 \sum_{\tau' \in \mathbb{T}^-} [\gamma_2 - f(\tau')]_+ \tag{4.6}$$

where $[.]_+ = \max(.,0), \gamma_1, \gamma_2 \in \mathbb{R}^+$. $\mathbb{T}^+, \mathbb{T}^-$ are the sets of positive and negative samples, and $\beta_1, \beta_2 > 0$ are constants denoting the importance of the positive and negative samples. This version of limit-based loss minimizes the aggregated error such that the score for the positive samples becomes less than γ_1 and the score for negative samples becomes greater than γ_2. To find the optimal limits for the limit-based loss, we suggest updating the limits during the training.

Time Complexity and Parameter Growth: Considering the ever growth of KGs and the expansion of the web, it is crucial that the time and memory complexity of a relational mode be minimal. Despite the limitations in expressivity, TransE is one of the popular models on large datasets due to its scalability. With $O(d)$ time complexity (of one mini-batch), where d is the size of embedding vectors, it is more effective than RESCAL, NTN, and the neural network models. Similar to TransE, the time complexity of MDE is $O(d)$. Due to the additive construction of MDE, the inclusion of more distance terms keeps the time complexity linear in the size of vector embeddings.

4.3 Model Extensions

4.3.1 Searching for the limits in the limit-based Loss

While the limit-based loss resolves the issue of margin ranking loss with distance-based embeddings, it does not provide a way to find the optimal limits. Therefore the mechanism to find limits for each dataset and hyper-parameter is the try and error. To address this issue, we suggest updating the limits in the limit-based loss function during the training iterations. We denote the moving-limit loss by $loss_{guide}$.

$$\begin{aligned} loss_{guide} = \lim_{\delta,\delta' \to \gamma_1} \beta_1 & \sum_{\tau \in \mathbb{T}^+} [f(\tau) - (\gamma_1 - \delta)]_+ \\ & + \beta_2 \sum_{\tau' \in \mathbb{T}^-} [(\gamma_2 - \delta') - f(\tau')]_+ \end{aligned} \tag{4.7}$$

where the initial value of δ, δ' is 0. In this formulation, we increase the δ, δ' toward γ_1 and γ_2 during the training iterations such that the error for positive samples minimizes as much as possible.

Instead of the fixed limit value as in limit-based loss, $loss_{guided}$ updates the limit for positive and negative objective functions during the training. The aim of this approach is to find a balance between two goals:

(i) To make the error of a correct triple near to zero, similar to the idea of margin ranking loss.

(ii) To increase the margin between the limits for positive and negative samples as much as possible.

The second goal is following the Structural risk minimization principle [83] to maximize the margin between the positive and negative samples. We minimize the limit for the objective of negative samples, with the condition that the error for the objective of positive samples stays small.

Therefore, we search for fine limits for the limit-based loss by testing on the validation set after each 50 epoch and taking those limits that give the best value during the tests. The details of the search for limits are explained in Algorithm 1.

This Algorithm impels the positive samples to gain zero loss(the idea of distance-based embeddings) and aims to increase a γ_2 as large as possible to maximize the margin between positive and negative loss. The Algorithm first sets the limit for positive samples and the limit for the negative samples second. After several iterations, if the positive loss ($loss^+$) does not decrease it implies that the limit for positive samples is too small. Therefore, it increases both γ_1, γ_2. Whenever during the iterations the $loss^+$ becomes zero it increase δ by a fixed amount ξ so that $\delta = \delta + \xi$.

After initialisation of the limits(γ_1 and γ_2) and δ and δ' it update the limits during the training iteration, and checks if $f(\tau) - f(\tau') \geq \gamma_2 - \gamma_1$ so that it preserve the characteristic of the margin-based ranking loss. It performs a similar comparison for the loss of negative values ($loss^-$) to decrease δ'.

After observing the most promising values for limits in the preset number of iterations, it stops the search and performs the training while having the δ values fixed(fixed limit-base loss) to allow the adaptive learning to reach loss values smaller than the $threshold$.

We based this approach on the idea of adaptive learning rate [23], where the Adadelta optimizer adapts the learning rate after each iteration, therefore in the $loss_{guided}$ we can update the limits without stopping the training iterations. In our experiments, the variables in the algorithm, are as

Algorithm 1 Guided Limit Loss

```
Initialize: $\delta = \delta' = \delta_0,\ \gamma_1 = \gamma_2 \in \mathbb{R}^+,\ \psi \in \mathbb{R}$
Initialize: $i = 0,\ \xi \in \mathbb{R}^+,\ threshold \in \mathbb{R}^+$
Inside training iterations:
if Using $loss_{guided}$ instead of $loss_{limit-based}$ then
    $loss^+ = \beta_1 \sum_{\tau \in \mathbb{T}^+} [f(\tau) - (\gamma_1 - \delta)]_+$
    $loss^- = \beta_2 \sum_{\tau' \in \mathbb{T}^-} [(\gamma_2 - \delta') - f(\tau')]_+$
    $loss\ =\ loss^+\ +\ loss^-$
    if $loss^+ = 0\ \ \&\ \ \gamma_1 \geq \xi$ then
        $\delta = \delta + \xi$
        if $loss^- > threshold\ \ \&\ \ \gamma_2 \geq \xi$ then
            $\delta' = \delta' + \xi$
if Using $loss_{limit-based}$ then
    $loss$ = the result from Equation (4.6)
```

follows: $\delta_0 = 0$, $threshold = 0.05$, $\xi = 0.1$.

We test the model after changes in δ and δ' and select those values that lead to the best ranking scores.

Lemma 3. There exist ψ and $\gamma_1, \gamma_2 \geq 0$ ($\gamma_1 \geq \gamma_2$), such that only if one of the terms in f_{MDE} estimates a fact as true, f_{MDE} also predicts it as a true fact. Consequently, the same also holds for the capability of MDE to allow learning of different relation patterns.

Proof. We show there are boundaries for $\gamma_1, \gamma_2, w_1, w_2, w_3, w_4$, such that learning a fact by one of the terms in f_{MDE} is enough to classify a fact correctly.

To extend this formulation to more term, it is enough to show that there is at least one set of boundaries for the positive and negative samples that follow the constraints.

The case to prove is when three of the distance functions classify a fact as negative N and the one distance function e.g. S_2 classifies it as positive P, and the case that S_1 and S_3 classify a fact as positive and S_2 classify it as negative. We set $w_1 = w_3 = 1/4$ and $w_2 = 1/2$ and assume that Sum is the value estimated by the score function of MDE, we have:

$$a > \frac{N}{2} \geq \frac{\gamma_2}{2} \wedge \frac{\gamma_1}{2} > \frac{P}{2} \geq 0 \Rightarrow a + \frac{\gamma_1}{2} > Sum + \psi \geq \frac{\gamma_2}{2} \tag{4.8}$$

There exist $a = 2$ and $\gamma_1 = \gamma_2 = 2$ and $\psi = 1$ that satisfy $\gamma_1 > Sum \geq 0$ and the inequality 4.8. □

It is notable that without the introduction of ψ and the limits γ_1, γ_2 from the limit-based loss, Lemma 3 does not hold, and framing the model with this settings makes the effective combination of the terms in f_{MDE} possible. In case future studies discover new interesting distances, this Lemma shows how to basically integrate them into MDE.

In contrast to SimplE, which ties the relation vectors of two terms in the score together, MDE does not directly relate them. This feature allows MDE to take advantage of independent relation and entity vectors and combine contrasting terms in the score function.

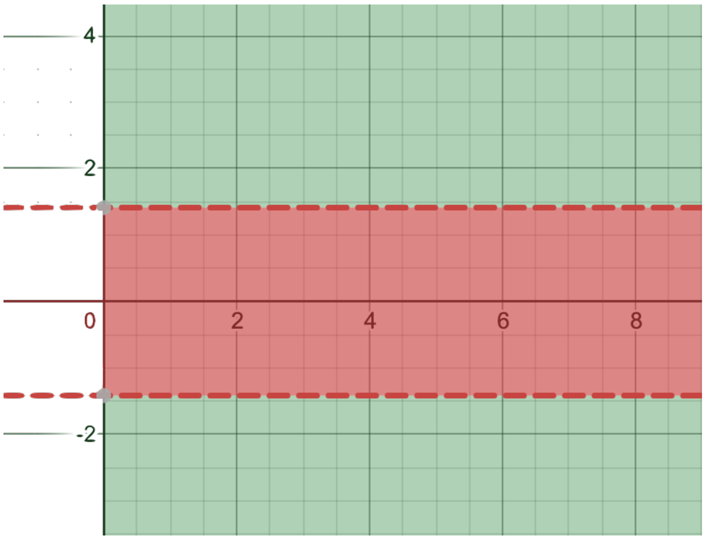


Figure 4.2: Illustration of the possible positioning of score values for MDE$_{NN}$ on WN18RR where the value of γ_1 and γ_2 is 2.

The learning of the symmetric relations is previously studied (e.g. in [12, 30]) and [84] studied the training over the inverse of relations, however providing a way to gather all these benefits in one model is a novelty of MDE. Besides, complementary modeling of different vector-based views of a knowledge graph is a novel contribution.

4.3.2 MDE$_{NN}$: MDE as a Neural Network

The score of MDE is already aggregating a multiplication of vectors to weights. We take advantage of this setting to model MDE as a layer of a neural network that allows learning the embedding vectors and multiplied weights jointly during the optimization. To create such a neural network we multiply ψ by a weight w_5 and we feed the MDE score to an activation function. We call this extension of MDE as MDE$_{NN}$:

$$\begin{aligned} f_{MDE_{NN}} = F(\parallel w_1 S_1^i \parallel_p + \parallel w_2 S_2^j \parallel_p + \parallel w_3 S_3^k \parallel_p \\ + \parallel w_4 S_4^l \parallel_p + \parallel w_5 \parallel_p c - \psi) \end{aligned} \tag{4.9}$$

where F is $Tanhshrink$ activation function with the formulation

$$Tanhshrink(x) = x - Tanh(x) \tag{4.10}$$

and $w_1, w_2, \ldots, w_5$ are elements of the latent vector w that are estimated during the training of the model and c and ψ are constants. Similarly we add y and z as latent vectors multiplied to the first and the second elements in the Equations 4.1, 4.2, 4.3 & 4.4. For example S$_1$ in MDE$_{NN}$ becomes:

$$S_1 = \parallel y_1 h + z_1 r - t \parallel_p \tag{4.11}$$

Dataset	#entity	#relation	#training	#validation	#test
FB15k	14 951	1 345	483 142	50 000	59 071
WN18	40 943	18	141 442	5 000	5 000
FB15k-237	14 541	237	272 115	17 535	20 466
WN18RR	40 943	11	86 835	3 034	3 134

Table 4.1: Number of entities, relations, and triples in each division.

This framing of MDE reduces the number of hyper parameters. In addition, the major advantage of MDE$_{NN}$ –in comparison to the linear combination of terms in MDE– is that the $Tanhshrink$ activation function allows the non-linear mappings between the embedding vectors and the expected target values for the loss function over positive and the negative samples.

Since $Tanhshrink$ has a range of $\mathbb{R}$ it allows setting large values for γ_1 and γ_2. For example, for WN18RR we set their value to 1.9. It is notable that the classic activation functions such as $sigmoid$ and $Tanh$ are not suitable to be used as activation functions here because they cannot converge the loss function to limit values larger than one. To generate a non-linear loss function for MDE$_{NN}$, we combine the square of positive loss and the negative loss values:

$$\begin{aligned} loss_{MDE_{NN}} &= (\sum_{\tau \in \mathbb{T}^+} [f(\tau) - \gamma_1]_+)^2 \\ &+ (\sum_{\tau' \in \mathbb{T}^-} [\gamma_2 - f(\tau')]_+)^2 \end{aligned} \tag{4.12}$$

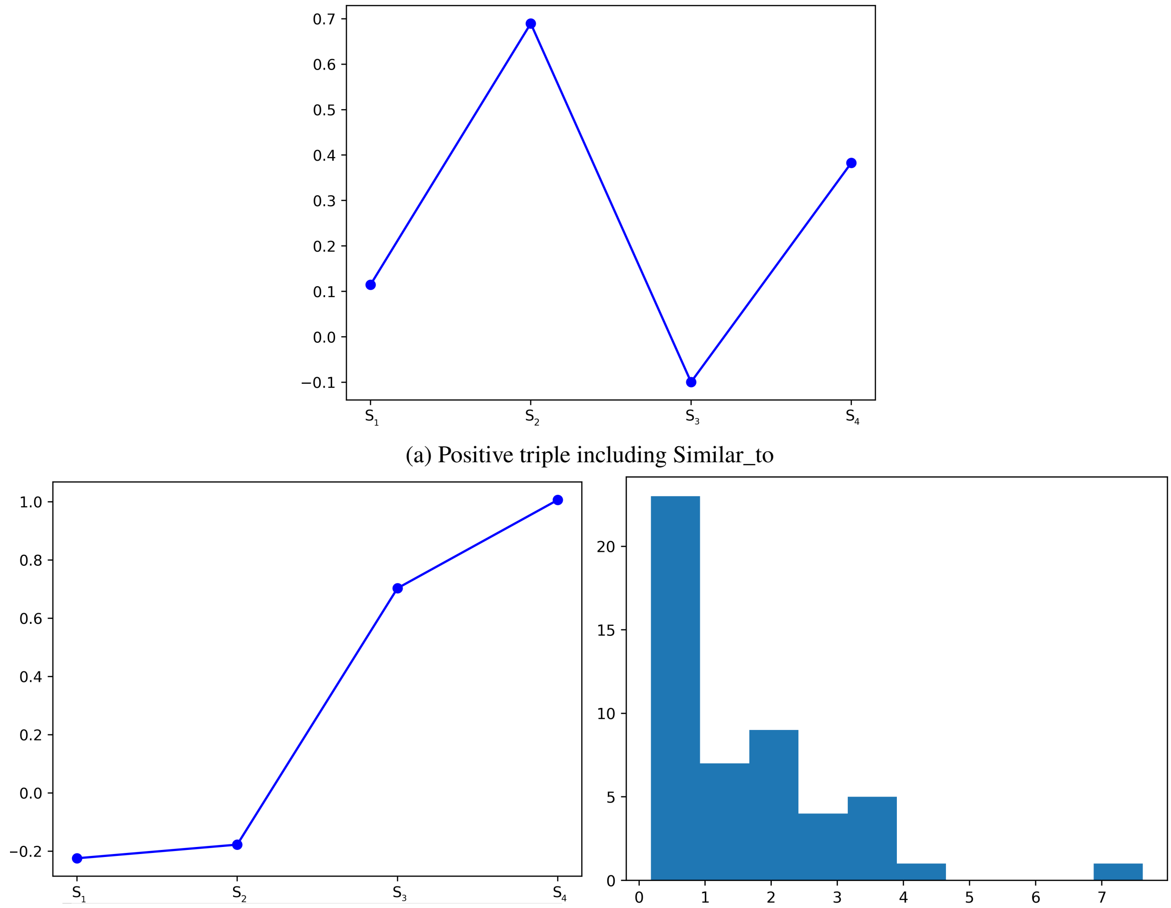
Figure 4.2 shows the positioning of the score values for MDE$_{NN}$ on WN18RR in which γ_1 and γ_2 is 2. The horizontal axis indicates the sample numbers and the vertical axis indicates their loss values. The score values for negative samples, $f(\tau')$ lay on the green area and score values for the positive samples, $f(\tau)$ lay on the red area.

4.4 Experiments

Datasets: We experimented on four standard datasets: WN18 and FB15k which were extracted by Bordes *et al.* in [14] from Wordnet [85] and Freebase [86] respectively. We used the same train/valid/test sets as in [14]. WN18 contains 40 943 entities, 18 relations and 141 442 train triples. FB15k contains 14 951 entities, 1 345 relations and 483 142 train triples. In order to test the expressiveness ability rather than relational pattern learning power of models, FB15k-237 [59] and WN18RR [47] exclude the triples with inverse relations from FB15k and WN18 which reduced the size of their training data to 56% and 61% respectively. Table 4.1 summarizes the statistics of these knowledge graphs.

Baselines: We compare MDE with several state-of-the-art relational learning approaches. Our baselines include TransE, RESCAL, DistMult, NTN, ER-MLP, ComplEx and SimplE. We report the results of TransE, DistMult, and ComplEx from [13] and the results of TransR and NTN from [87], and ER-MLP from [74]. The results on the inverse relation excluded datasets are from Table 13 of [30] for both TransE and RotatE. And the rest are from [47][3].

[3] Scores of ConvE on FB15k are from https://github.com/TimDettmers/ConvE/issues/26



(a) Positive triple including Similar_to

(b) Negative triple including symmetric pattern Similar_to

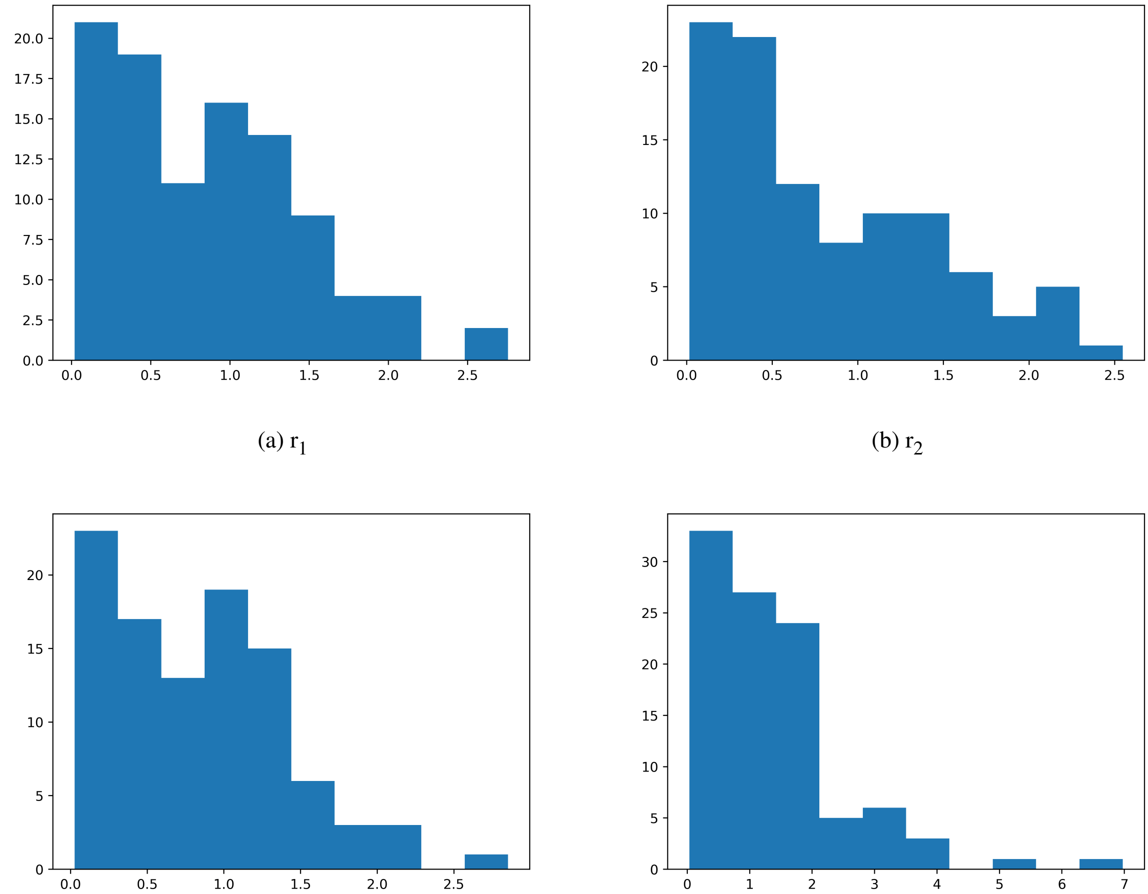
(c) hypernym + hyponym

Figure 4.3: Diagrams describing the prediction of each term in MDE score for symmetric relation in a positive triple are depicted in Figure (a) and its corrupted version with the same head and tail in Figure (b). Figure (c) shows the histogram diagram of the elements of two the sum of two inverse relations, hypernym and hyponym in S_1.

Evaluation Setting: We evaluate the link prediction performance by ranking the score of each test triple against its versions with replaced head, and once for the tail. Then we compute the hit at N (Hit@N), mean rank (MR), and mean reciprocal rank (MRR) of these rankings. We report the evaluations in the filtered setting.

Implementation: We implemented MDE in PyTorch[4]. Following [79], we generated one negative example per positive example for all the datasets. We used Adadelta [23] as the optimizer and fine-tuned the hyperparameters on the validation dataset. The ranges of the hyperparameters are set as follows: embedding dimensions 25, 50, 100, 200, batch size in the range of 1024 to 1725, and iterations 50, 100, 1000, 1500, 2500, and 3600. We set the initial learning rate on all datasets to 10. For MDE, the best embedding size and γ_1 and γ_2 and β_1 and β_2 values on WN18 were 50 and 1.9, 1.9, 2 and 1 respectively and for FB15k were 200, 10, 13, 1, 1.

[4] `https://pytorch.org`



(a) r_1

(b) r_2

(c) r_3

(d) r_1+r_2-r_3

Figure 4.4: Diagrams (a, b, c & d) show the norm of the elements in vectors r_1, r_2, r_3 and r_1+r_2-r_3 where r_3 is composed of r_1 and r_2. Here, r_1 represents /award/award_category/nominees./award/ award_nominatio/nominated_for , r_2 represents /award/award_nominee/award_nominations./award /award_nomination/nominated_for and r_3 represents /award/award_winner/awards_won./award/ award_honor/award_winner.

The best found embedding size and γ_1 and γ_2 and β_1 and β_2 values on FB15k-237 were 100, 9, 9, 1 and 1 respectively and for WN18RR were 50, 2, 2, 5 and 1. We selected the coefficient of terms in (4.5), by grid search, with the condition that they make a convex combination, in the range 0.1 to 1.0, and tested those combinations of the coefficients where they create a convex combination. Found values are $w_1 = 0.16$, $w_2 = 0.33$, $w_3 = 0.16$, w_4=0.33. We experimented the model to find the best value for ψ between $\{0.1, 0.2, \ldots, 1.5\}$. We use $\psi = 1.2$ for the MDE experiments. We use the value 2 for p in p-norm throughout the paper.

To regulate the loss function and to avoid over-fitting, we estimate the score function for two sets of independent vectors and we take their average in the prediction. Another advantage of this operation is the reduction of required training iterations.

	Countries(AUC-PR)		
Model	S1	S2	S3
DistMult [12]	**1.00** ± **0.00**	0.72 ± 0.12	0.52 ± 0.07
ComplEx [13]	0.97 ± 0.02	0.57 ± 0.10	0.43 ± 0.07
ConvE [47]	**1.00** ± **0.00**	0.99 ± 0.01	0.86 ± 0.05
RotatE [30]	**1.00** ± **0.00**	**1.00** ± **0.00**	0.95 ± 0.00
MDE	**1.00** ± **0.00**	**1.00** ± **0.00**	**1.00** ± **0.00**

Table 4.2: Results on the Countries datasets. Results of RotatE are taken from [30] and the results of the other models are from [47].

.

For the WN18RR experiment of MDE$_{NN}$, we use the same parameters as in MDE for γ_1, γ_2, and the same embedding size. We use the embedding size 50 for WN18RR, 200 for WN18, 200 for FB15k-237, and 200 for FB15k. We use c = 4 and $\psi = 2.5$ for the MDE$_{NN}$ experiments. Our experiments show this method usually reaches its best performance in the benchmarks in just 50 iterations.In our experiments, we use an adaptive learning rate method for both MDE and MDE$_{NN}$.

We observe that adding new three dimensions and three distance functions to the model has not slowed the model's training. In contrast, it allowed the model to reach 0.92 in just 100 epochs with a training time of 8 minutes when running on a system with a Pentium i5 CPU.

The current framework of KG embedding model evaluations is based on the open-world assumption that the generation of an unlimited number of negative samples is possible. In this setting, it becomes debatable to consider negative sample generation as a part of the model since it significantly influences the ranking results. In particular, RotatE effectively assimilates the effect of many negative samples in the self-adversarial negative sampling technique. We verify the influence of this sampling method on the MDE results and to distinguish it we call this implementation MDE$_{adv}$. For this implementation, we use Adam as the optimizer similar to RotatE. We select dimension 400, learning rate 0.0005, batch size 512, and 624 negative samples per positive sample for the test on WN18RR. For FB15k-237, we test the model with dimension 1000, learning rate 0.0005, batch size 240, and 1224 negative samples per positive sample.

4.4.1 Relation Pattern Implicit Inference

To verify the implicit learning of relation patterns, we evaluate our model on Countries dataset [73, 74]. This dataset is curated in order to explicitly assess the ability of the link prediction models for composition pattern modeling and implicit inference. It is made from 2 relations and 272 entities, where the entities include 244 countries, 5 regions, and 23 subregions. In comparison to general link prediction tasks on knowledge graphs, evaluation queries in Countries are specified only to the form locatedIn(c, ?), where, the answer is one of the five regions. The Countries dataset is made of 3 tasks, and each one requires inferring a composition pattern with increasing length and difficulty. The measure for this evaluation is usually AUC-PR.

Table 4.2, shows that our model performs significantly better than the previous models. While RotatE outperforms older models on S1 and S2, MDE gains the best result on S1 and S2 as well as S3, which is the most difficult task. We also evaluate if MDE embeddings implicitly represent different relation patters.

	WN18			FB15k		
Model	MR	MRR	Hit@10	MR	MRR	Hit@10
TransE [14]	–	0.454	0.934	–	0.380	0.641
TransH [29]	303	–	0.867	87	–	0.644
STransE [71]	206	0.657	0.934	69	0.543	0.797
RESCAL [22]	–	0.890	0.928	–	0.354	0.587
DistMult [12]	–	0.822	0.936	–	0.654	0.824
SimplE [72]	–	0.942	0.947	–	0.727	0.838
NTN[37]	–	0.53	0.661	–	0.25	0.414
ER-MLP [36]	–	0.712	0.863	–	0.288	0.501
ConvE [47]	504	0.942	0.955	51	0.657	0.831
ComplEx [13]	–	0.941	0.947	–	0.692	0.84
RotatE [30]	309	**0.949**	**0.959**	**40**	**0.797**	**0.884**
MDE	**118**	0.871	0.956	49	0.652	0.857

Table 4.3: Results on WN18 and FB15k. Best results are in bold.

Symmetry pattern requires S_3 term to correctly distinguish positive and negative samples for MDE. We investigate the relation embeddings from a 50-dimensional MDE trained on WN18. Figure 4.3a gives the value of different terms for a triple with symmetric relation “similar_to” between the entities “pointed” and “sharpened”. Since the smaller score values of MDE are suggesting that a triple is a positive sample, the smaller values of individual terms in the model would also influence the overall model to recognize a triple as positive. S_3 shows the smallest value among all the terms. Figure 4.3b illustrates the values of terms for the negative sample (pointed, similar_to, pointed) where S_1 and S_2 scores are low due to their incapability in recognizing a negative sample when the head and tail are the same. However, S_3 adjusts the overall MDE score by producing a great number that compensates for the low S_1 and S_2 results.

Inversion pattern requires inverse relations in S_1 and S_2 terms to have inverse angles. Figure 4.3c shows the histogram of the elements of the sum of hypernym and hyponym relations in S_1. We can see from this Figure that most of the elements in these two relations have opposite values.

Composition pattern requires the embedding vectors of the composed relation to be the addition of the other two relations in S_1. We train a 200-dimensional MDE model to verify the implicit inference of the composition patterns on FB15k-237. Figure 4.4a to 4.4d illustrate that most of the elements in $r_1 + r_2 - r_3$ are near zero where r_3 is composed of r_1 and r_2 relations.

4.4.2 Link Prediction Results

Table 4.3 summarizes our results on FB15k and WN18. It shows that MDE performs like RotatE and outperforms other state-of-the-art models in MR and Hit@10 tests, significantly improving the performance of the latent distance approaches. Table 4.4 shows the results of the experiments on FB15k-237 and WN18RR. These results follow the same pattern as the ones reported in Table 4.3. This Table shows the extension of the model with adversarial negative sampling gives the best MRR result in the FB15k-237. The slight improvement of MDE_{NN} over MDE in hit@10 ranking results demonstrates the positive effect of the non-linear setting for limits of the loss function and the non-linear activation function.

	WN18RR			**FB15k-237**		
Model	MR	MRR	Hit@10	MR	MRR	Hit@10
DistMult [12]	5110	0.43	0.49	254	0.241	0.419
ComplEx [13]	5261	0.44	0.51	339	0.247	0.428
ConvE [47]	5277	0.46	0.48	246	0.316	0.491
RotatE [30]	3340	**0.476**	**0.571**	**177**	0.338	**0.533**
MDE	**2629**	0.457	0.536	189	0.288	0.484
MDE$_{NN}$	3165	0.432	0.531	-	-	-
MDE$_{adv}$	3219	0.458	0.560	203	**0.344**	0.531

Table 4.4: Results on WN18RR and FB15k-237. Best ones are in bold.

The significantly large Hit@10 rank value and the small mean rank value indicates overfitting in the model. By using limited-based loss, and fixing the limits for loss of positive and negative functions to a constant we reduce the mean rank to a great extent. We observe that the mean rank in MDE is for WN18 is the best value and in FB15k is lower than most of the other methods.

It is noticeable that the addition of independent vectors in the model does not decrease the mean rank of the model, whereas, in models with high vector dimensions, the MR and MRR results are unbalanced. For example, for ComplEx and ConvE which both use a vector dimension of 200, the MRR is significant but the MR is high (which is not suitable), such that, CompEx and ConvE have the greatest overfitting among the compared methods. A reason besides the higher dimension, for their overfitting, is that their training method includes an unbalanced positive and negative sample batch in the loss calculation. On a different note, RotatE mitigates this issue with the application of a high number of negative samples per positive sample, which allows for having higher dimensions, and balancing out the effect of the greater number of negative samples by normalizing the loss of negative samples in comparison to the number of positive samples in a batch.

The comparison of our model to other state-of-the-art methods in Table 4.4, shows the competitive performance of MDE and MDE$_{adv}$. It is observable that in the MDE tests with only one negative sample per positive sample and using vector sizes between 50 to 200, MDE challenges models with relatively large embedding dimensions (1000) and a high number of negative samples (up to 1024). In the ablation study presented in [30], we notice that RotatE (with the margin-based ranking criterion, and without self-adversarial negative sampling) produces a Hit@10 score of 0.476 on FB15k-237, which is lower than the MDE score.

The adaptation of self-adversarial negative sampling in MDE improves the Hit@10 ranking and the MRR score of the model. This improvement is more significant on the FB15k-237 rather than on the WN18RR, as there is a greater number of relations and entities in FB15k-237 and the self-adversarial negative sampling increases the coverage of different combinations of entities in the training. We also observe on the FB15k-237 benchmark, that MDE$_{adv}$ outperforms previous models on the MRR score since it exists more relations with composition pattern in this dataset than in the WN18RR dataset.

We include each of the terms in MDE as we hypothesize that each one contributes to the generalization power of the model. Practically, we verify this approach in the following section.

	WN18RR			FB15k-237		
Individual Term	MR	MRR	Hit@10	MR	MRR	Hit@10
S_1	3137	0.184	0.447	187	0.260	0.454
S_2	8063	0.283	0.376	439	0.204	0.342
S_3	3153	0.183	0.449	**186**	0.258	0.455
S_4	**2245**	**0.323**	**0.467**	220	**0.273**	**0.462**

Table 4.5: Results of each individual term in MDE on WN18RR and FB15k-237. The best results are in bold.

	WN18RR			WIN18		
Removed Term	MR	MRR	Hit@10	MR	MRR	Hit@10
S_1	3983	0.417	**0.501**	**113**	0.838	**0.946**
S_2	**3727**	0.358	0.490	131	0.823	0.943
S_3	3960	0.427	0.499	161	0.850	0.943
S_4	3921	0.366	0.478	163	0.705	0.929
None	3985	**0.428**	**0.501**	151	**0.844**	**0.946**

Table 4.6: Results of MDE after 100 iterations when removing one of the terms. The best results are in bold.

4.4.3 Ablation Study

To better understand the role of each term in the score function of MDE stated in Equation (4.5), we embark on two ablation experiments. First, we train MDE using one of the terms alone and observe the link prediction performance of each term in the filtered setting. In the second experiment, we remove one of the terms at a time and test the effect of this removal on the performance model after 100 training iterations. We perform the first experiment on WN18RR and FB15k-237 and the second on WN18RR and WN18 datasets.

Table 4.5 summarizes the results of the first experiment on WN18RR and FB15k-237. We can see that S_4 outperforms the other terms while S_1 and S_3 perform very similarly on these two datasets. Among the four terms, S_2 performs the worst since most of the relations in the test datasets follow an anti-symmetric pattern, where S_2 is not effective in modeling them. Comparing each term to the overall MDE model shows that none perform as well as the MDE combination. In addition, simply aggregating their MRR produces a greater number than MDE, which indicate the leakage of relation patterns in WN18RR and FB15k-237 datasets.

Table 4.6 shows the results of the second experiment. The evaluations on WN18RR and WN18 show that the removal of S_4 has the most negative effect on the performance of MDE. The removal of S_1, which was one of the good performing terms in the last experiment, has the most neglectable effect. This result indicates that while S_1 covers some of the relation patterns that are targeted in other terms, it is not as effective as the combination of these terms. Nevertheless, S_1 improves the MRR in the MDE. Also, when we remove S_2, the MRR and Hit@10 are negatively influenced, indicating that it exists cases where S_2 performs better than the other terms. However, it performed the worst among all the terms in the individual tests.

4.5 Conclusion

In this Chapter, we created a model based on the generation of several independent vectors for each entity and relation that overrides the expressiveness restrictions of most of the embedding models. To our knowledge, MDE is one of the few existing KG embedding approaches that are unable to allow the modeling of all the three very frequent relation patterns. We framed MDE into a Neural Network structure and validated our contributions via both theoretical proofs and empirical results.

We demonstrated that with multiple views to translation embeddings and by using independent vectors (it was previously supposed to cause poor performance [77, 72]), a model can perform solidly in the link prediction task. Our experimental results confirm the competitive performances of MDE in MR and Hit@10 on the benchmark datasets. Particularly, MDE outperforms all the current state-of-the-art models for the benchmark of composition relation patterns.

CHAPTER 5

Graph Feature Aware Knowledge Graph Embedding

Knowledge graph embedding (KGE) is lately at the center of many artificial intelligence studies due to its applicability for solving downstream tasks, including link prediction and node classification. To target the KGE challenge more effectively, in the previous Chapter, we proposed a multi-objective method that allows learning relation patterns of knowledge graphs. The training mechanism of MDE, as in most knowledge graph embedding models, encodes node relations into the vector space, utilizing only the local graph structure of an entity, i.e., information of the 1-hop neighborhood. However, capturing global features of entities besides the local graph structure is crucial for prediction tasks on knowledge graphs.

This Chapter proposes a novel KGE method named Graph Feature Attentive Neural Network (GFA-NN) that extracts graphical features of entities. GFA-NN not only considers the local graph structure of a knowledge graph as in the conventional methods but also generates embeddings that entail the graph features. Consequently, the resulting embeddings of GFA-NN are attentive to two types of global network features; First, nodes' relative centrality, based on the observation that some of the entities are more "prominent" than the others, and second, the relative position of entities in the graph. GFA-NN computes several centrality values per entity, generates a random set of reference nodes' entities, and computes a given entity's shortest path to each entity in the reference set. It then learns this information through optimization of objectives specified on each of these features.

Besides theoretically analyzing the proposed graph feature learning, we empirically investigate the performance of GFA-NN on several link prediction benchmarks. We show that GFA-NN achieves on-par or better results than state-of-the-art KGE solutions in both inductive and transductive settings. In addition, we investigate the potential of GFA-NN in embedding large knowledge graphs by testing it on a large-scale biological dataset, where it significantly outperforms the current models. In this Chapter, we target the second research question:

Research Question 2

Does learning network features of knowledge graphs improve the efficiency of KG embedding?

Contributions of this Chapter are as follows:

- Proposing a score function for embedding graph features alongside relational learning.
- Providing the mathematical formulation of several graph features, normalizing and adopting these features to make them compatible with relational learning.
- Proposing GFA-NN model: a neural network for feature-aware knowledge graph embedding.
- Providing a theoretical analysis for learning graph features in GFA-NN, highlighting the potential and limitations.
- Empirically evaluating graph feature-aware embedding of knowledge graphs, where it outperforms the state-of-the-art KGE methods in several benchmarks.

This Chapter is based on the following publication [27]:

- **Afshin Sadeghi**, Diego Collarana, Damien Graux, Jens Lehmann. *Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) 2021, 548–564.

Knowledge graphs (KGs) are capable of integrating heterogeneous data sources under the same graph data model. Thus KGs are at the center of many artificial intelligence studies. KG nodes represent concepts (entities), and labeled edges represent the relation between these entities, e.g. (Berlin, CapitalOf, Germany) is a fact stating Berlin is the capital of Germany. KGs such as Wikidata, WordNet, Freebase, and Nell include millions of entities and relations representing the current knowledge about the world. KGs, in combination with Machine Learning models, are used for refining the knowledge graph itself and for downstream tasks, like link prediction and node classification. However, to use KGs in Machine Learning methods, we need to transform the graph into vector space representations named knowledge graph embeddings (KGE).

KGE has many applications, including analysis of social networks and biological pathways. Thus, many approaches have been proposed ranging from translation methods, e.g., Trans* family [14, 28, 29]; Rotation-based methods, e.g., RotatE [30]; Graph Convolutional methods, e.g., R-GCN [88], CompGCN [44], and TransGCN [89]; and Walk-based methods , e.g., RDF2Vec [90].

Traditional graph embedding methods, however, rely exclusively on facts (triples) that are explicitly present in a knowledge graph. Therefore, their prediction ability is limited to a set of incomplete facts. A means of improvement is to incorporate complementary information in the embeddings. A class of methods applies external knowledge such as entity text descriptions [91] and text associations related to entities[92] into the KG modeling. In contrast, intrinsic methods extract complementary knowledge from the same KG. For example, the algorithms that derive logical rules from a KG and combine them with embeddings of the KG [93, 94]. Analogously recent studies [49] consider graph structural features as an intrinsic aspect of KGs in the embedding.

This Chapter addresses a challenge of most KGE models; These methods independently learn the existence of relation from an entity to its hop-1 neighborhood. This learning strategy neglects the fact that entities located at a distance can still affect an entity's role in the graph. Besides that, the location of the entities in the network can be beneficial for distinguishing nodes.

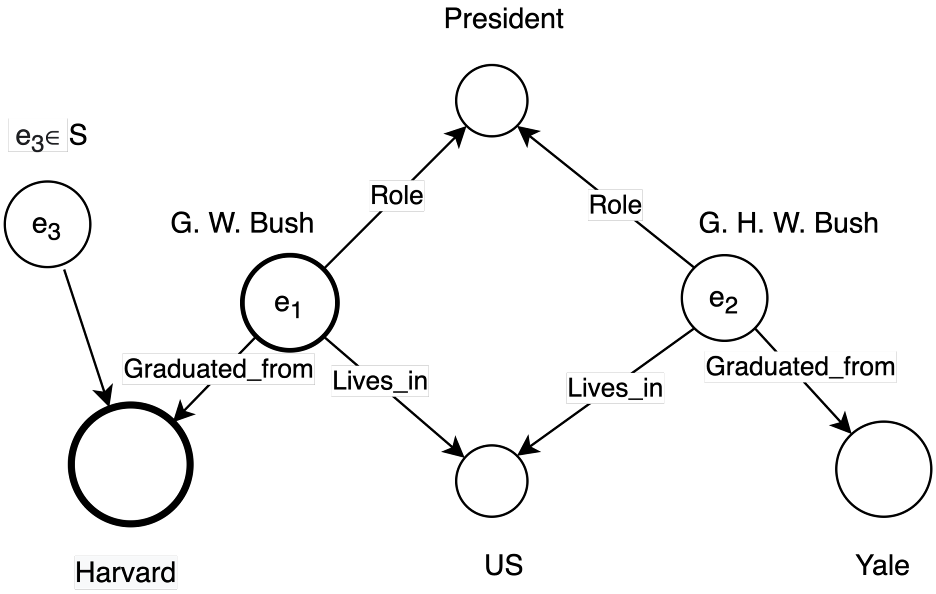


Figure 5.1: An example knowledge graph in which nodes e_1 and e_2 are difficult to distinguish by a KGE model only using their neighborhood information.

Figure 5.1 illustrates such an example where the goal is to learn embeddings for e_1 and e_2 entities in the KG. Distinguishing between the two candidates, i.e., George W. Bush and George H. W. Bush, is challenging for previous methods since e_1 and e_2 have almost the same neighbors, except George W. Bush graduated from Harvard University while George H. W. Bush did not.

However, If we compare e_1 to e_2 using their eigenvector centrality, we can easily distinguish them. e_1 has a greater centrality than e_2 since e_1 is connected to Harvard that has a high eigenvector centrality. Analogously, if we consider the shortest path of e_1 and e_2 to e_3 that belongs to set of reference node S, their distance to e_3 is different. Intuitively, if a model could beforehand know the centrality and distance to e_3 as additional knowledge, it can more easily model e_1 and e_2 and rank them correctly.

With a new view to knowledge graph embedding, we propose GFA-NN[1], an approach that learns both the local relations between the entities and their global properties in one model. In order to effectively encode entity indicators in knowledge graph modeling, we focus on learning node centrality and positional indicators (e.g., the degree, Katz, or eigenvalue centrality of entities in the graph) as well as the knowledge graph structure.

For this purpose, we fuse the modeling of each entity indicator in the style of Multiple Distance Embedding (MDE) [15] where distinct views to knowledge graphs are modeled through independent embedding weights.

GFA-NN extracts positional information and four centrality indicators of nodes from the KG and defines a learning function for each one. Then GFA-NN scores their aggregation with MDE. Previously, different leanings were applied to embedding models using constraints in the loss function. Now that MDE has broken the limitation of using more than one objective function on independent embeddings, we directly add new extracted information about the entities as aggregated objective functions.

[1] Source code is available at `https://github.com/afshinsadeghi/GFA-NN`

Centrality values and the position of nodes in graphs are global measurements for nodes across the whole graph. If we use a local assignment, for example, the number of paths between specific nodes, this measurement may have different weights based on what portion of the network is considered in the calculation.

Despite the exciting recent advancements, most of the previous works fail to learn the relation between entities regarding the whole graph. Therefore, we define relative position attentive and relative centrality attentive functions for embedding the relative importance of nodes and their position relative to the whole network. In the following section, we discuss the relation between our work and the current state-of-the-art.

We outline in Section 5.2 the idea of centrality and positional qualities learning and explain our approach. In Section 5.3, we mention the model's theoretical analysis; and we continue with experiments that evaluate our model in Section 5.4.

5.1 Background

We conceive a KG as a multi-relational graph. An entity in such formulation is equivalent to a node in graph theory, and an edge represents a relation. In this study, we use Node and Entity interchangeably. We use the term “Node" to emphasize its graphical properties. and we use the term “Entity" to highlight the entity's concept.

Link prediction on knowledge graphs is made by a Siamese classifier that embeds KG's entities and relations into a low-dimensional space. Thus, a knowledge graph embedding model is a function $f : \mathcal{E}, \mathcal{R} \rightarrow \mathcal{Z}$, that maps entities $\mathcal{E}$ and relations $\mathcal{R}$ to d-dimensional vectors $\mathcal{Z} = \{z_1, \ldots, z_n\}$, $z_i \in \mathbb{R}$.

In the following, we first review the definition for Structure-based Embedding from the Chapter 2 and then provide a generalization of the position-aware embedding definition [49] that distinguishes our method from the previous works:

Structure-based Embedding: A KG embedding $z_i = f : \mathcal{E}, \mathcal{R} \rightarrow \mathcal{Z}$ is attentive to network structure if it is a function of entities and relations such that it models the existence of a neighborhood of an entity e$_i$ using relations r$_i$ and other entities $e_j \in \mathcal{E}$. Most knowledge graph embedding methods like QuatE and RotatE compute embeddings using the information describing connections between entities and, therefore, structure-based.

Property-Attentive Embedding: A KG embedding $z_i = f : \mathcal{E}, \mathcal{R} \rightarrow \mathcal{Z}$ is attentive to network properties of an entity if there exists a function $g_p(., ., ...)$ such that $d_p(v_i, v_j, ...) = g_p(z_i, z_j)$, where $d_p(,)$ is a graphical property in G. This definition includes both the property of a sole node such as its centrality and the properties that describe the inter-relation of two nodes such as their shortest path. Examples of Property-Attentive Embedding are P-GNNs [49] and RDF2Vec [90], which their objective function incorporates the shortest path between nodes into embedding computation.

We show that current KGE methods cannot recover global graph properties, such as path distances between entities and centrality of nodes, limiting the performance in tasks where such information is beneficial. Principally, structure-aware embeddings cannot be mapped to property-aware embeddings. Therefore, only using structure-aware embeddings as input is not sufficient when the learning task requires node property information. This work focuses on learning KGEs capturing both entities' local network structures conjointly with the global network properties. We validate our hypothesis that a trait between local and global network features is crucial for link prediction and node classification tasks.

A KGE is attentive to node network properties if the embedding of two entities and their relation can be used to approximately estimate their network feature, e.g., their degree relative to other entities in the network.

You *et al.* [49] show for position attentive networks, there exists a mapping g that maps structure-based embeddings $f_{st}(v_i), \forall\, v_i \in$ V to position attentive embeddings $f_p(v_i), \forall\, v_i \in V$, if and only if no pair of nodes have isomorphic local q-hop neighborhood graphs. This proposition justifies the good performance of KGE models in tasks requiring graphical properties and their under-performance in real-world graphs such as biological and omniscience KGs (e.g., Freebase, DBpedia), in which the structure of local neighborhoods are quite common. This proposition, however, does not hold for centrality attentive embeddings. The reason is that if no pair of nodes have isomorphic local q-hop neighborhood graphs, it is still possible for them to have the same centrally attentive embeddings. For example, two nodes with the same number of neighbors consisting of different nodes have the same degree; however, their neighborhoods are non-isometric. We formulate this proposition below:

Proposition:

For position attentive networks, we know (You *et al.* [49]) that there exists a mapping g that maps structure-based embeddings $f_s q(v_i), \forall\, v_i \in$ V to position attentive embeddings $f_p(v_i), \forall\, v_i \in V$, if and only if no pair of nodes have isomorphic local q-hop neighbourhood graphs.
This proposition does not hold for centrality attentive embeddings.

Proof. If no pair of nodes have isomorphic local q-hop neighborhood graphs, it is still possible for them to have the same centrally attentive embeddings. To show that, it is enough that two nodes have the same centrally value in the graph: for example, for degree centrality, when the two nodes have the same number of neighbors, that are consisting of different nodes, their neighborhoods are non-isometric; however, they have the same degree centrality.

Therefore $f_p(v_i)$ can not be a function of $f_{st}(v_i)$, and the two pairs of nodes would have different structure-aware node embeddings while their centrally attentive embeddings are equal. □

We show in Section 5.2 how we address this challenge for centrality learning.

5.2 Method

This Section details our proposed method for generating entity network properties attentive embeddings from knowledge graphs. We generalize the concept of knowledge graph embedding with a primary insight that incorporating centrality and distance values enables KGE models to compute embeddings with respect to the graphical proprieties of entities relative to the whole network instead of only considering the direct local neighbors (Figure 5.2, left side).

When modeling the positional information, instead of letting each entity model the information independently and selecting a new reference set per iteration, we keep a set of reference entities through training iterations and across all the networks in order to create comparable embeddings. This design choice enables the model to learn the position of nodes with respect to the spectrum of different reference node positions and makes each embedding attentive to position (Figure 5.2, top left).

GFA-NN models each graphical feature with a dedicated objective function, meaning that the information encrypted in centrality attentive embeddings does not interfere with the embedding vectors

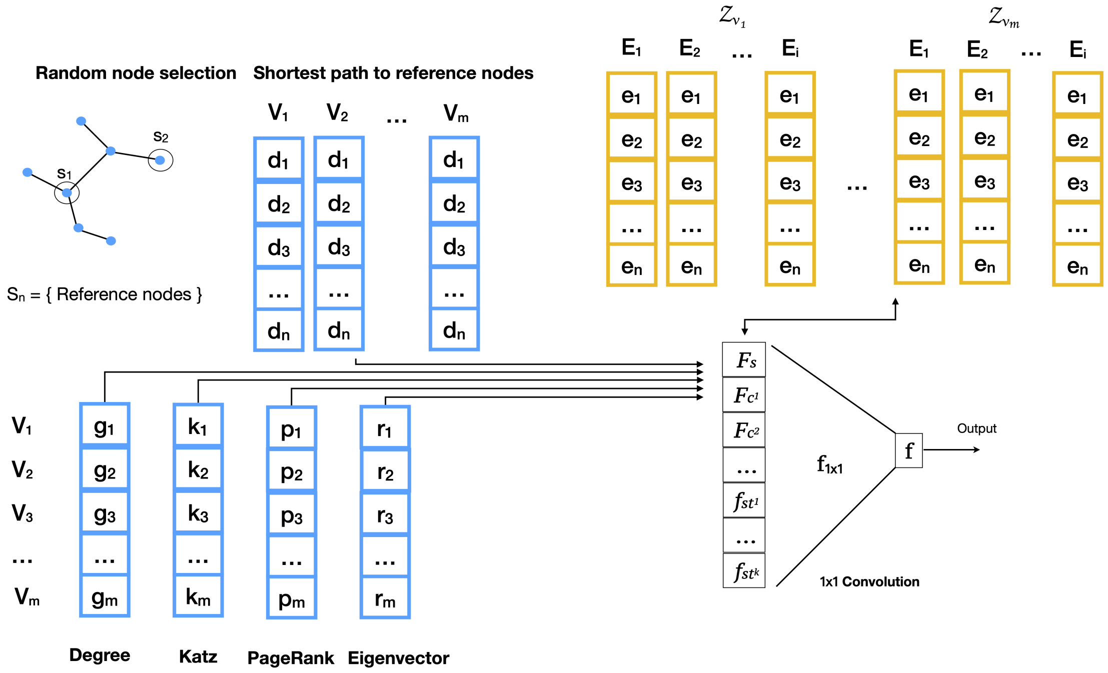


Figure 5.2: Architecture of GFA-NN. GFA-NN first pre-computes the centrality property of nodes and their distance to a set of to randomly selected reference nodes (**Left**). Then, node centrality and position embeddings attentive to position z_{v_m} are computed via scores $F_1, ..., F_k$ from the distance between a given node v_i and the reference-sets S_i which are shared across all the entities (**Top-middle**).

that keep the positional information (Figure 5.2, top right). To compute the embedding z_{v_1} for node v_1, a score of GFA-NN first computes via function F_i and then aggregates the F_i scores via 1×1 convolution and an activation function over obtains a vector of final scores. Inside 1×1 a vector w learned, which is used to reduce scores into one centrality and position-aware score and produces embeddings z_{v_1} which is the output of the GFA-NN (Figure 5.2, right side).

Centrality for nodes are individual values. While positional values are calculated relative to a set of nodes in a graph, only one centrality value per entity is extracted. Still, learning this information is valuable because the centrality value of a node is meaningful despite the absence of a large portion of the network. This trait is particularly beneficial in inductive relation prediction tasks.

5.2.1 Model Formulation

The components of GFA-NN are as follows:

- Random set of reference nodes for distance calculations.
- Matrix M of distances to random entities, where each row i is a set of the shortest distance of an entity to the selected set of random nodes.
- Trainable vectors r_d, h_d, t_d that project distance matrix M to a lower dimensional embedding space $z \in \mathcal{R}_k$.
- Structure-attentive objective functions $f_{st^1}(v_i), \ldots, f_{st^k}(v_i)$ that model the relatedness information of two entities with their local network, which is indicated by triples that consist of head and tail nodes (entities) connected by an edge (relation).

- Position-attentive objective function F_s that models the position of a node (entity) in the graph with respect to its distance to other nodes. This objective considers these distances as a factor of the relatedness of entities.
- Centrality attentive objective functions F_c that model the relatedness information of two entities according to centrality properties of nodes (entities). In this setting, the global importance of nodes is learned relative to the centrality of other nodes.
- Trainable aggregation function $f_{1\times 1}$ is a 1×1 convolution [95] that fuses the modeling of the structure-based connectivity information of the entities and relations with their position aware and centrality attentive scoring.

Our approach consists of several centrality and position-attentive phases that each of which learns an indicator in a different metric of the status of entities relative to the network.

In the first phase, GFA-NN performs two types of computation to determine the position status and the centrality status of entities. The unit for centrality status computes the relative significance of entities as a vector of length one c_i^j, where j represents each of the centrality metrics. The unit for position status embedding samples n random reference-entities S_n, and computes an embedding for entities. Each dimension i of the embedding is obtained by a function F that computes the shortest path to the i-th reference entity relative to the maximum shortest path in the network.

Then objective functions $F_s, F_c^1, ..., F_c^4$ apply an entity interaction model to enforce the property features e_i^s into entity embeddings e_i, which in the next phase makes a 1×1 convolution [95] over the scores via weights $w \in \mathbb{R}^r$ and non-linear transformation *Tanhshrink*. Specifically, each entity earns an embedding per attribute that includes values that reveal the relative status information from input entity network properties information. Calculation of the centrality for all nodes in the network leads to a vector representation of the graph for each measure, while the distances to the reference nodes S generate a dense matrix representation.

The network property attentive modeling functions are the same class of functions as used by existing translational KGEs plus a modeling function of embeddings that we extended to be performed in 3D using a rotation matrix. In the following, we further elaborate on the design choices.

5.2.2 Centrality-Attentive embedding:

As shown in Section 5.1, the centrality values are not canonical. Therefore, the model learns their difference in a normal form, in which the equality of their norm does not mean they are equal. Degree centrality is defined as : $C_d(n) = \deg(n)$.

Katz centrality [96] extends degree centrality from counting neighbor nodes to nodes that can be connected through a path, where the contribution of distant nodes is reduced:

$$C_k(n) = \sum_{k=1}^{\infty} \sum_{j=1}^{N} \alpha^k A_{j,i}^k$$

where A is the adjacency matrix and α is attenuation factor in the range $(0, 1)$. Another included centrality measure is PageRank with the following formulation:

$$C_p(n) = \alpha \sum_j a_{j,i} \frac{C_p(j)}{L(j)} + \frac{1 - \alpha}{N}$$

where N is $|V|$, the number of nodes in the graph, and $L(j)$ is the degree of node j. Relative eingenvector centrality score of a node n is defined as:

$$C_e i(n) = \frac{1}{\lambda} \sum_{m \in KG} a_{m,n} x_m$$

where $A = (a_{v,t})$ is the adjacency matrix such that $a_{v,t} = 1$ if node n is linked to node m, and $a_{v,t} = 0$ otherwise. λ is a constant which fulfils the eingenvector formulation $Ax = \lambda x$. Note that the method in first phase normalizes each of the centrality values. The normalization occurs with respect to minimum and the maximum value for nodes in the network and makes attributes relative to the whole network. For example, degree centrality is normalized as follows:

$$C_i^d = \frac{degree(i) - degree_{min}}{degree_{max} - degree_{min}}$$

The centrality-attentive modeling embeddings functions are the same class of dissimilarity functions used by existing KGEs plus a penalty we define on the difference of the entity embeddings as:

$$F_{c^d} = \|h_i - t_i\|_2 \quad - \quad \|\cos(log(C_h^d)) - \cos(log(C_t^d))\|_2 \tag{5.1}$$

where the function is normalized with the l_2 norm, h_i and t_i represent the vector representation of head and tail in a triple and lastly, C_h^d and C_t^d respectively denote the centrality values of the head and tail entities in that triple.

5.2.3 Position-Attentive embedding:

GFA-NN models the neighborhood structure using rotations in 3D space and a penalty that forces the method to encode the difference in distances of entities to the reference nodes. The formulation for the structure-attentive part is:

$$F_{rot} = \| v_h - v_r \otimes v_t \|_2 \tag{5.2}$$

where $\otimes$ represents a rotation using a rotation matrix of Euler angles with the formulation of direction cosine matrix (DCM):

$$\begin{bmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\phi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{5.3}$$

where ϕ, θ ,and ψ are Euler angles. The modeling of positional information is performed by a score function made from rotation matrices and a penalty:

$$F_p = F_{rot} - \| \cos(S_i^h) - \cos(S_i^t)) \|_2 \tag{5.4}$$

where S_C^i is the calculated distance from the head and tail nodes to the reference nodes. Hence, the score enforces to learn structure-attentive embeddings with a penalty that is the normalized scalar difference of distance to reference nodes. Here we use the l_2 norm to regularize the F_i score functions and apply adversarial negative sampling technique to generate weighted scores per sample [30]. We utilise Adam [24] for optimization.

Reference-set selection relies on a Gaussian random number generator to select normally distributed random reference nodes from the network. GFA-NN keeps a fixed set of reference nodes during the training of different entities through different iterations to generate embeddings attentive to the position that are in the same space and, hence, comparable to each other.

Multiple Property aware scores can be naturally fused to achieve higher expressive power. This happens in $f_{1\times1}$. Since canonical position-attentive embeddings do not exist, GFA-NN also computes structure-attentive embeddings h_v via the common distance-based modelings of MDE. These scores are aggregated with attribute attentive scores, and then the model using a linear combination of these scores forms a 1×1 convolution to produce only one value that contains both properties. The output of this layer is then fed into the nonlinear activation function.

It is notable that independent weights in MDE formulation allow restricting solution space without limiting the learnability power of the model. Note also that the method is still Semi-supervised learning, where the train and test data are disjoint, and the centrality and path information computation does not consider the portion of the unknown network to the model and only exist in the test data.

5.3 Theoretical analysis

5.3.1 Connection to Preceding KGE Methods

GFA-NN generalizes the existing knowledge graph embedding models. Taking the definition for the structure-aware and node properties attentive models into perspective, existing knowledge embedding models use the same information of connecting entities through different relations techniques but use different neighborhood selection scoring functions and sampling strategies, and they only output the structure-aware embeddings.

GFA-NN shares the score function aggregate training with MDE [15]. There, a linear combination of scores $f_{1\times1} = \sum w_i F_i$ is trained, where w_i weights are learnt together with the embeddings in the score functions F_i. GFA-NN also shares the concept of training independent embeddings with MDE. The direction cosine matrix used in modeling positional information is convertible into a four-element unit quaternion vector (q_0, q_1, q_2, q_3). The quaternions are the center of the structure-based model QuatE [17], where the relations are models as rotations in the quaternion space. Here, besides modeling rotation, we formulated the score to include a translation as well. RotatE [30] similarly, formulates the relations with a rotation and reduction in $\parallel v_h \circ v_r - v_t \parallel$, however RotatE models rotation in the complex space. In the branch of Graph neural networks, the aggregate information of a node's neighborhood in one-hop [97, 98, 44] or nodes in the higher hops [99] is used in the message passing mechanism.

P-GNN [49] explicitly learns the shortest path of random nodes for simple graphs. However, it takes a new set of reference nodes in each iteration, which makes the learning of shortest paths local and incremental. In addition, it makes it difficult to retain the structural information from positional embedding. GFA-NN generalizes positional learning by learning the distances to a fixed set of random nodes through the whole network, which makes the positional embedding vectors globally comparable. From the point of view of graph type, GFA-NN generalizes the positional learning to multi-relational graphs to support KGs. GFA-NN not only learns a weight for each of the network features, but it also associates it with the existing relation types between the two entities that their features are being learned. By including the relation type into position-attentive embeddings, the position also is encoded into relation vectors that connect the entities.

Note that relation type learning is sub-optimal for learning centrality values because the dimension of relation types is much higher than the dimension of the node property values (one integer value), which makes the centrality value differentiation diminish when learned together with the association information belonging to relations. Another aspect that GFA-NN generalizes the existing graph learning algorithms is that this method learns several centrality aspects and positional information at the same time.

5.3.2 Expressive Power

In this Section, we explain how GFA-NN generalizes the expressive power of knowledge graph embedding methods from the perspective of a broader Inductive bias. Generally, inductive bias in a learning algorithm allows it to better prioritize one solution over another, independent of the observed data [100].

Assuming that a labeling function y labels a triple (h, r, t) as $d_y^r(h, t)$, we predict y^r, similar to [49] from the perspective of representation learning, which is by learning an embedding function f, where $v_h = f(v, G)$ and f computes the entity embeddings for v_h, v_r and v_t. Thus, the objective becomes the task of maximizing the probability of the conditional distribution $p(y|v_h, v_r, v_t)$. This probability can be designated by a distance function $d_v(v_h, v_r, v_t)$ in the embedding space, which usually is an l_p norm of the objective function of the model.

A KGE model, with a goal to predict the existence of an unseen triple (h, r, t), learns embeddings weights v_h and v_t for the entities h and t and v_r for a relation r that lies between them. In this formulation, the embedding for an entity e is computed based on its connection through its one-hop neighborhood, which we express by structural information S_e, and optimization over the objective function $f_\theta(e, S_e)$. Hereby, the neighborhood information of two entities S_{e_1} and S_{e_2} is computed independently. However, the network feature attentive objective function f_ϕ in GFA-NN poses a more general inductive bias that takes in the distance from a random shared set of reference nodes, which are common across all entities, and the centrality values, which are relative to all nodes. In this setting, any pair of entity embeddings are correlated through the reference-set and the spectrum of relative centrality and therefore are not independent anymore. We call this feature attentive information I.

Accordingly, we define a joint distribution $p(w_{e_1}, w_{e_2})$ over node embeddings, where $w_{e_i} = f_\phi(e_i, I)$. We formalize the problem of KG representation learning by minimizing the expected value of the likelihood of the objective function in margin-based ranking setting, in the following for a structure base KGE:

$$\begin{aligned}\min_\theta \, & \mathbb{E}_{e_1,e_2,e_3,S_{e_1},S_{e_2},S_{e_3}} \\ & \mathcal{L}(d_v^+(f_\theta(e_1, S_{e_1}), f_\theta(e_2, S_{e_2})) - d_v^-(f_\theta(e_1, S_{e_1}), f_\theta(e_3, S_{e_3})) - m)\end{aligned} \tag{5.5}$$

and in GFA-NN:

$$\min_\theta \mathbb{E}_{e_1,e_2,e_3,I} \quad \mathcal{L}(d_v^+(f_\phi(e_1, I), f_\phi(e_2, I)) - d_v^-(f_\phi(e_1, I), f_\phi(e_3, I)) - m) \tag{5.6}$$

where d_v^+ is the similarity metric determined by the objective function for a positive triple, indicating existing a predicate between entities and by optimizing converges to the target label function $d_y(e_1, e_2) = 0$ for positive samples(existing triples) and $d_y(e_1, e_3) = m$ on negative samples. Here, m is the margin value in the margin ranking loss optimization setting.

Table 5.1: Statistics of the data sets used in the Experiments.

Dataset	#entities	#relations	#train	#validation	#test
WN18RR	40943	11	86835	3034	3134
FB15k-237	14541	237	272115	17535	20466
ogbl-biokg	45085	51	4762678	162886	162870
WN18RR-v_3-ind	5084	11	6327	538	605
WN18RR-v_4-ind	7084	9	12334	1394	1429
NELL-995-v_1-ind	225	14	833	101	100
NELL-995-v_4-ind	2795	61	7073	716	731

Note that the representations of entities are calculated using joint and marginal distributions, respectively. Similar to the proof of expressive power in [49], considering the selection of entities $e_1, ..., e_i \in G$ as random variables to form any triples, the mutual information between the joint distribution of entity embeddings and any $Y = d_y(e_1, e_2)$ is more significant than that between the marginal distributions.

$Y : I(Y; X_{joint}) \geq I(Y; X_{marginal})$. Where,
$X_{joint} = (f_\phi(e_1, S_{e_1}), f_\phi(e_2, S_{e_2})) \sim p(f_\phi(e_1, S_{e_1}), f_\phi(e_2, S_{e_2}))$
$X_{marginal} = (f_\theta(e_1, I), f_\theta(e_2, I))$

Because the gap of mutual information is large when the targeted task is related to positional and centrality information of the network, we deduce that KGE embedding based on the joint distribution of distances to reference nodes and relative centrality values has more expressive power than the current structure-based KGE models.

5.3.3 Complexity Analysis

Next, we explain the complexity of the method and show its complexity compared to the structure-based models. When the shortest paths are calculated on the fly, the learning complexity is added up by $O(b\ log(b))$ for finding the shortest paths on b entities in each batch, and similarly, the centrality computation aggregates to the complexity. We, therefore, pre-calculate this information to separate them from the learning complexity. The complexity of each of the objective functions on a batch with size b is $O(b)$, and suppose n property attentive features and m structure-aware scores be involved, the overall complexity becomes $O((n+m)\ b)$. Note that the larger number here is b, and the complexity increases by b times when a graphical feature is involved in the learning.

5.4 Experiments

We evaluate the performance of our model with three experiments. The first experiment is the traditional transductive ranking benchmark on datasets extended from the work initially introduced in [14]. The second experiment is another transductive ranking evaluation on a large-scale dataset, and the third is the inductive relation prediction experiment. This experiment evaluates a model's ability to generalize the link prediction task to unseen entities in the inductive setting. Table 5.1 shows the statistics of the datasets used in the experiments.

Table 5.2: Results on WN18RR and FB15k-237. Best results are in bold.

	WN18RR			**FB15k-237**		
Model	MR	MRR	Hit@10	MR	MRR	Hit@10
ComplEx-N3	–	0.48	0.57	–	0.37	0.56
QuatE[2]	–	0.482	0.572	–	**0.366**	**0.556**
TuckER	–	0.470	0.526	–	0.358	0.544
CompGCN	3533	0.479	0.546	197	0.355	0.535
RotatE	3340	0.476	0.571	177	0.338	0.533
MDE	3219	0.458	0.536	203	0.344	0.531
GFA-NN	3390	**0.486**	**0.575**	186	0.338	0.522

Metrics and Implementation: We evaluate the link prediction performance by ranking the score of each test triple against all possible derivable negative samples by once replacing its head with all entities and once by replacing its tail. We then calculate the hit at N (Hit@N), mean rank (MR), and mean reciprocal rank (MRR) of these rankings. We report the evaluations in the filtered setting. We determine the hyper-parameters by using a grid search. We select the testing models that give the best validation set results. We generally fix the learning rate on 0.0005 and search the embedding size amongst {200, 300, 400, 500}. We search the batch size from {250, 300, 500, 800, 1000}, and the number of negative samples amongst {10, 100, 200, 400, 600, 800, 1000}. We describe all GFA-NN hyper-parameters in the end of this Section[2].

5.4.1 Transductive link prediction experiment

Datasets: We perform experiments on three benchmark datasets: WN18RR [47], FB15k-237 [59], and ogbl-biokg [60], which is comparably a sizeable knowledge graph assembled from a large number of biomedical repositories.

Baselines: We compare our model with several state-of-the-art structure-based embedding approaches. Our baselines include RotatE [30], TuckER [101], ComplEx-N3 [102], QuatE [17], MDE [15] and the recent graph neural network CompGCN [44]. We report results of each method on WN18RR and FB15k-237 from their respective papers, while the results of the models in ogbl-biokg are from [60]. For RotatE, we report its best results with self-adversarial negative sampling, and for QuatE, we report the results with N3 regularization. Our model uses the exact self-adversarial negative sampling introduced in RotatE. This negative sampling schema is also applied to all the other models in the ogbl-biokg benchmark.

Results and Discussion: Table 5.2 and Table 5.3 summarize the performance of GFA-NN and other KGE models in the transductive link prediction task. We observe that GFA-NN outperforms other state-of-the-art KGEs on WN18RR and is producing competitive results on FB15k-237.

Our analysis shows that the standard deviation of different positional and centrality measures through the network in WN18RR is ≈0.009, while in FB15k-237, it is ≈0.002, which is 4.5 times smaller. This comparison indicates that in WN18RR, these features are more diversified, but in FB15k-237, they are close to each other. This analysis suggests the crucial impact of learning centrality and positional-attentive embeddings on the superiority of the GFA-NN on the WN18RR benchmark.

[2] As well in the source code's manual: https://github.com/afshinsadeghi/GFA-NN

Table 5.3: MRR Results for `ogbl-biokg`. (Results of previous models are from [60].)

Method	Validation	Test
TransE	0.7456	0.7452
DistMult	0.8055	0.8043
ComplEx	0.8105	0.8095
RotatE	0.7997	0.7989
GFA-NN	**0.9011**	**0.9011**

Table 5.4: Hit@10 results for inductive datasets. (Other models' results are from [51].)

Model	WN18RR-v_3-ind	WN18RR-v_4-ind	NELL-995-v_1-ind	NELL-995-v_4-ind
NeuralLP	0.4618	0.6713	0.4078	**0.8058**
DRUM	0.4618	0.6713	0.5950	**0.8058**
RuleN	0.5339	0.7159	0.5950	0.6135
GraiL	0.5843	0.7341	0.5950	0.7319
GFA-NN	**0.5893**	**0.7355**	**0.9500**	0.7722

While the result on the FB15k-237 is still very competitive to the state-of-the-art, as a lesson learned, we can declare it as a fixed procedure to perform the standard deviation analysis on a dataset before determining how much the network property attentive embedding learning method would be beneficial.

Table 5.3 shows the **MRR** evaluation results on the comparably large biological dataset named as `ogbl-biokg`. In this benchmark, the number of entity and training samples is much larger than the WN18RR and FB15k-237 datasets.

The capability of learning feature attentive embeddings is crucial in this transductive link prediction task. While the best KGEs can only achieve the MRR of 0.8105 on the validation and 0.8095 on the test dataset, GFA-NN reaches 0.901 on both datasets, improving state-of-the-art by 9 percent. This wide gap between the results supports the assumption that property-attentive embeddings surpass prior methods in larger-scale real-world networks. A cause for this significant efficiency difference is that traditional methods have difficulty differentiating entities in large-scale networks due to the greater number of similar substructures. In contrast, GFA-NN can distinguish such nodes better by learning the global features of nodes. In addition, in such a small-world structured network, the entity-to-relation ratio is more substantial, which causes a considerable standard deviation of positional and centrality qualities. As indicated earlier, this feature is beneficial to the model's efficiency.

5.4.2 Inductive link prediction experiment

Datasets: For evaluations in the inductive setting, we select four variant datasets which Komal et al. [51] extracted from WN18RR and NELL-995 [62].

Baselines: Inductive baselines include GralL [51], which uses sub-graph reasoning for inductive link prediction. RuleN [103] that applies a statistical rule mining method, and two differentiable methods of rule learning NeuralLP [104] and DRUM [105]. We report the results of these state-of-the-art models from Komal et al. [51].

Table 5.5: Best hyperparameter setting of GPA-NN on the benchmark datasets.

Dataset	**Dim**	**Batch size**	**#neg**	**#iterations**	**g value**
WN18RR	400	300	800	100000	2.5
FB15k-237	1000	1000	200	200000	4.0
ogbl-biokg	400	600	850	700000	2.5
WN18RR-v_3-ind	300	1000	200	30000	0.5
WN18RR-v_4-ind	200	1400	10	30000	- 0.5
NELL-995-v_1-ind	200	300	600	20000	2.5
NELL-995-v_4-ind	200	1000	700	20000	2.5

Results: Table 5.4 summarizes the GFA-NN's Hit@10 ranking performance against methods specified on the inductive link prediction task. Although we did not explicitly design GFA-NN for this task, we observe that GFA-NN performs very competitively in this setting and outperforms the best inductive learning models in most cases. This result supports our hypothesis that the knowledge graph embedding attentive to positional and centrality qualities are beneficial for prediction tasks in challenging settings, i.e., inductive link prediction tasks.

Hyperparameter Settings: We list the best hyperparameters setting of GPA-NN on the benchmark datasets in Table 5.5. The learning rate in all the experiments is fixed to 0.0005, adversarial temperature for negative sampling is fixed to 2.5, and ψ, the dividend for the score aggregation in $f_{1\times1}$ is fixed to 14.

5.5 Conclusion

In this Chapter, with a new view of the relational learning algorithms, we proposed to learn the structural information of the network conjointly with learning the centrality and positional properties of the knowledge graph entities in one model. We provided theoretical analyses and empirical evaluations to identify the improvements and constraints in the expressive power for this class of KGEs. In particular, we demonstrated that with proper formulation, learning these global features is beneficial to the link prediction task, given that GFA-NN performs highly effectively in various benchmarks and often outperforms current state-of-the-art solutions in both inductive and transductive settings. We showed this approach can be put forth for tasks on large-scale graphs. Since GFA-NN is effective on networks with a higher entity-to-relation ratio, applications of the approach can be considered on biological, chemical, and social networks in future works.

CHAPTER 6

BenchEmbed: A Framework for Reproducible Link Prediction Benchmarking

In the previous Chapters 4 and 5, we proposed methods for effective embedding of knowledge graphs. To effectively compare KGE methods in the link prediction task, testing them on their best parameters under FAIR [106] conditions is crucial. In this Chapter, we target a common challenge in the evaluations of the knowledge graph embedding methods, i.e., the lack of reproducible evaluations. Specifically, a framework to generate reproducible experiments that follow the FAIR principles is missing.

In this study, we extend the general HOBBIT benchmarking platform to evaluate the link prediction efficiency of embedding models with such criteria. The source code of this study, a demonstrating benchmark made by the framework, and the installation and usage guide of the framework are openly available in `https://github.com/mlwin-de/BenchEmbed`.

This Chapter targets the research question:

Research Question 3

How can we make the evaluations of embedding models reproducible?

Contributions of this Chapter can be summarized as follows:

- Developing the Benchembed framework that generates fixed test environments to perform reproducible link prediction evaluation of KGE models.
- Presenting the framework structure and providing a base for understanding the approach.
- Providing instructions for adaptation of the work in further studies to promote the application of reproducible link prediction experiments in future works.
- Providing templates to simplify the extension and deployment of the framework to promote reproductive link prediction studies.

This Chapter is based on following publication [107]:

- **Afshin Sadeghi**, Xhulia Shahini, Martin Schmitz, Jens Lehmann *BenchEmbedd: A FAIR Benchmarking tool for Knowledge Graph Embeddings*. Demo track SEMANTiCS 2021. Xhulia Shahini and Martin Schmitz collaborated on this paper's related code implementation and deployment tests.

A gap in current KGE studies is a standard independent evaluation environment that evaluates the efficiency of models in the FAIR setting (e.g., with the exact vector sizes). Furthermore, these studies suffer from the lack of a systematic, reproducible evaluation. To target these issues, we extended the HOBBIT [108] platform as a Holistic benchmarking approach for Big Linked Data. With a new set of benchmarks to evaluate the efficiency of knowledge graph embedding models with the aforementioned criteria. We dedicate the rest of this Chapter to explaining the structure of this Benchmarking tool and demonstrating the usage of the benchmarking system for the knowledge graph embedding models.

We chose HOBBIT as the base because it is developed under FAIR principles [106]. We follow the same concepts in making this BenchEmbedd, which are:

1. F1 (Meta) data are assigned a globally unique and persistent identifier.

2. F2 Data are described with rich metadata (defined by R1 below).

3. F3 Metadata clearly and explicitly includes the identifier of the described data.

4. F4 (Meta) data are registered or indexed in a searchable resource.

Another advantage of the platform is generating dockerized benchmarking, i.e., once a system (image) is generated, it is executed locally on a personal computer or a local cluster or deployed on computing services such as Amazon Web Services (AWS).

The produced benchmarks are accessible, transferable, and easily reusable. This setting promotes reliable scientific publications because it allows researchers to repeat the evaluations of a study without concerns about standardized evaluation hardware. We ensure the reproducibility of the evaluations by generating independent benchmark units that are executable (docker) images of the exact environment of an initial evaluation made by a researcher. The framework is easily extensible and allows adding more models.

In the following Section, we explain the structure of our benchmarking platform. We then present the functionalities of the framework in Section 6.2 and Section 6.3 illustrates a demonstration of BenchEmbedd and explains the steps to make a new test unit, i.e., Benchmark System.

6.1 Structure:

Figure[1] 6.1 illustrates the components in the HOBBIT platform structure. To make a HOBBIT-based benchmark, we created the green and orange components in this figure. These parts consist of the Benchmark Components (in orange) and Benchmark System (in green).

[1] The HoBBIT platform structure diagram is from [108].

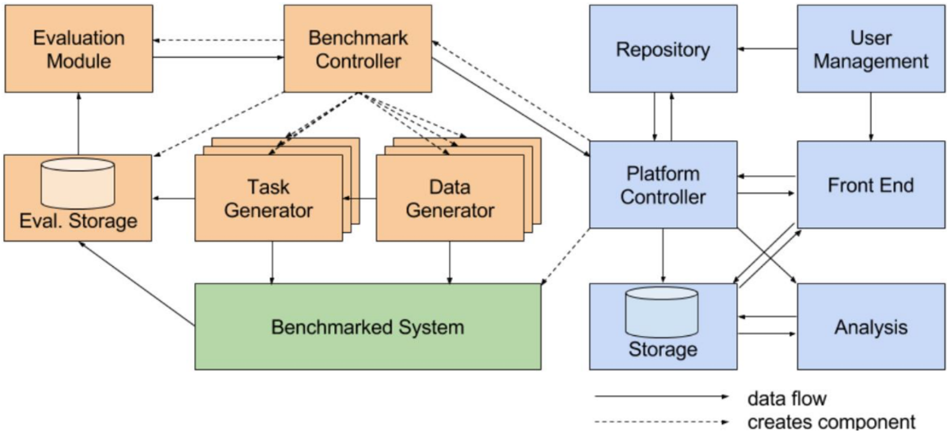


Figure 6.1: HOBBIT platform structure. BenchEmbedd extends it with evaluations and metrics for Link prediction on knowledge graphs.

The **Benchmark Components** provide the tasks and data for the experiment unit (system). The benchmark components work together as an infrastructure for benchmarking an experiment unit in a link prediction experiment. This section consists of Evaluation Module, Evaluation storage, Benchmark Controller, Task Generator, and Data Generator.

The **Benchmark System** contains a complete ready-to-run Benchmarking workflow within a controlled dockerized[2] running environment. A Benchmark System can contain configurations for running multiple tests on different models and datasets. The benchmarking platform includes a System configuration template with different configurations. It is possible to extend the framework to include different test datasets by editing the template.

6.2 Functionalities

A unit of BenchEmbedd performs a concrete Link Prediction evaluation experiment. While KGE models learn knowledge graphs in triples (head, relation, tail), the link prediction task tests KGE models on how effectively they predict missing links (triples) in a knowledge graph.

Figure 6.2 shows a knowledge graph with four entities, where the green relations are known. In this example, the link prediction task tests how well the missing triple (“Polito”, “is a university in”, “Italy”) is estimated by a knowledge graph learning model. A KGE model is effective if it generates a high score for the missing link indicating the existence of this relation. The current implementation computes the following metrics: HIT@1, HIT@3, HIT@10, and Mean Reciprocal Rank. Our published framework includes the test for TransE [14] model, and it is open to extending to other models. We configured a benchmark to test over the WN18RR benchmarking dataset for the demo.

[2] https://www.docker.com

6.3 Demonstration

The benchmark is a Java Maven project. After the setup [3] of BenchEmbedd, to execute a sample Benchmark system, online one needs to follow these steps:

1. Login to the website `https://master.project-hobbit.eu/`.
2. Select "Benchmarks".
3. Select "MLwin Benchmark" in the drop-down list of "Benchmarks".
4. Select the desired System to Benchmark in the drop-down list "System".
5. Press the "Submit" button.

At this stage, a popup window will appear. The Experiment Status shows the progress of the running experiment, and clicking the link in the popup window displays the experiment results once the experiment is finished. Figure 6.3 illustrates an example of the result table after running the demo benchmark system.

6.4 Defining new Benchmark Environments:

When Benchembedd generates a benchmark unit, it is not changeable anymore. Therefore, we generate a new Benchmark test unit for each test the requires different metrics or datasets. On our GitHub page, we provide a template Benchmark configuration, with the name Benchmark System file, that includes instructions necessary to make a new Benchmark environment. It is possible to add new tests and datasets by setting their associated variables in this file. In the following, we explain the steps to define a new benchmark environment using the extended template Benchmark configuration.

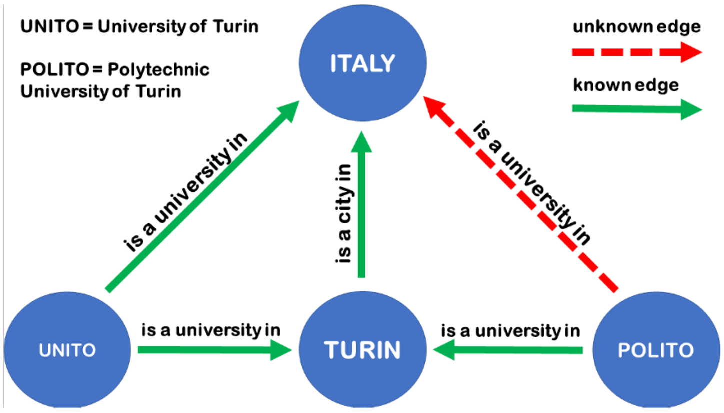
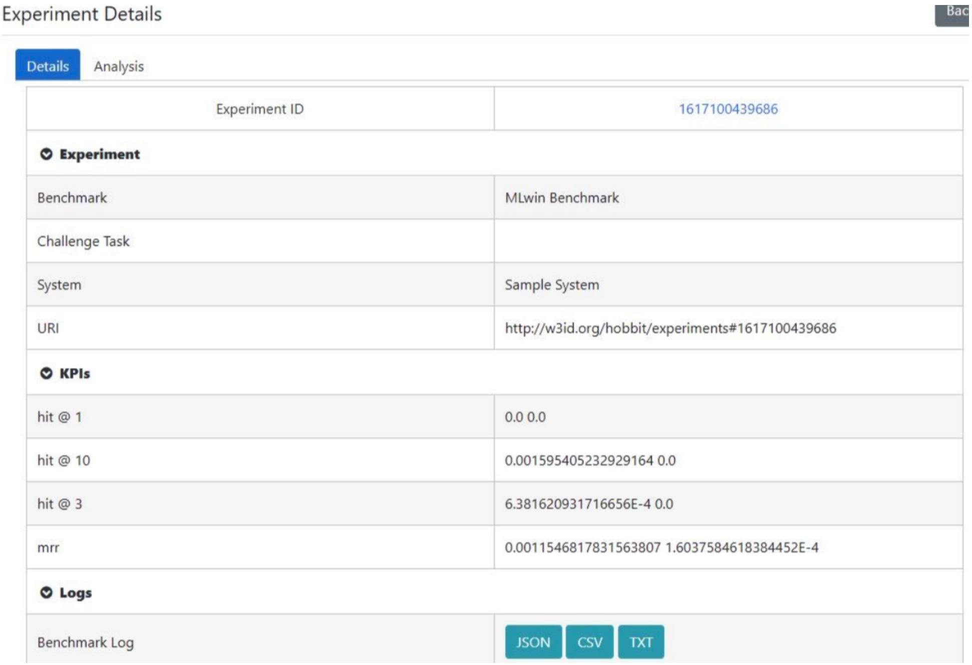


Figure 6.2: An example of a knowledge graph with a missing link.

[3] Setup guide is in `https://github.com/mlwin-de/BenchEmbedd#installation`



Experiment ID	1617100439686
Experiment	
Benchmark	MLwin Benchmark
Challenge Task	
System	Sample System
URI	http://w3id.org/hobbit/experiments#1617100439686
KPIs	
hit @ 1	0.0 0.0
hit @ 10	0.001595405232929164 0.0
hit @ 3	6.381620931716656E-4 0.0
mrr	0.0011546817831563807 1.6037584618384452E-4
Logs	
Benchmark Log	JSON CSV TXT

Figure 6.3: An example of demonstrated evaluation results.

The benchmark generation steps includes:

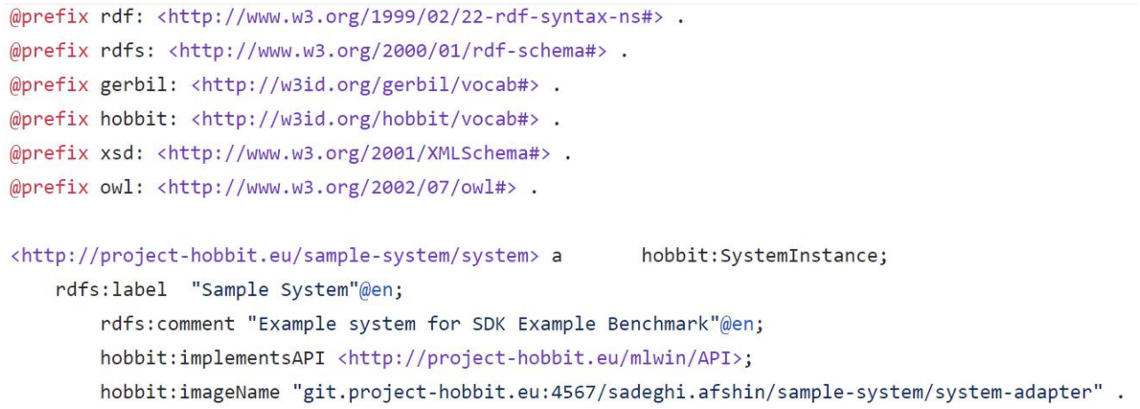
1. Writing a Benchmark System file.
2. Providing the set of pre-trained embedding vectors.
3. Creating a system docker image.
4. Writing a system meta-data file.
5. Creating a HOBBIT GitLab account to load up the files.

The steps to write a Benchmark System file are:

1. Extend the TransEtest.java file for a new benchmark system file. It contains the method “test_triple” that is the base for the link prediction tests.
2. Provide trained embeddings with names “entity2vec.txt” and “relation2vec.txt”.

Figure 6.1 shows a new independent Benchmark unit (dockerized system) with the color green and names it “Benchmarked System”. To illustrate the framework in action, have set up and published an executable Sample Benchmark unit (system) [4]. That dockerized unit contains the implemented framework and the trained embedding vector files with the TransE model. The output files of the training process for this repository are converted from “.npy” to “.txt” files using the script located at

[4] mvn commands: https://github.com/mlwin-de/BenchEmbedd#benchmark-the-system-online



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix gerbil: <http://w3id.org/gerbil/vocab#> .
@prefix hobbit: <http://w3id.org/hobbit/vocab#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://project-hobbit.eu/sample-system/system> a       hobbit:SystemInstance;
    rdfs:label  "Sample System"@en;
        rdfs:comment "Example system for SDK Example Benchmark"@en;
        hobbit:implementsAPI <http://project-hobbit.eu/mlwin/API>;
        hobbit:imageName "git.project-hobbit.eu:4567/sadeghi.afshin/sample-system/system-adapter" .
```

Figure 6.4: An example of system meta-data file.

“src/kge_output_to_data.py”.

HOBBIT project allows publication and deployment of benchmarks generated by Benchembedd. To deploy and upload the Benchembedd unit in the HOBBIT GitLab, a user requires a HOBBIT GitLab account. A user can create an account for a new deployment in `git.project-hobbit.eu`.

HOBBIT GitLab recognizes a Benchembedd user name and a system name of a test unit by reading a meta-data file inside a test unit. We included a file “system.ttl” that provides a template to declare this information. After setting this information, a user can push a test unit system to HOBBIT GitLab as a docker image. Figure 6.4 shows an example of a system meta-data file whose label is adopted to “sample-system” and includes a GitLab username.

6.5 Conclusion

This Chapter presents Benchembedd, a framework that generates fixed test environments to perform reproducible link prediction experiments for KGE models. In order to facilitate the research on Link Prediction and KG embeddings, we developed this framework while keeping in mind the ease of reproducible benchmark creation. To support further Benchmarks in the link prediction domain, besides explaining the framework’s structure, we illustrated the method of adaptation and deployment of the benchmarking system.

CHAPTER 7

Relational Pattern Benchmarking

The previous Chapters 4 and 5 proposed approaches that target different challenges in the representation learning of knowledge graphs. The most common procedure to assess the efficiency of the knowledge graph embedding methods is to put them under the Link Prediction empirical analysis. The Link Prediction based on KG embeddings targets the sparsity and incompleteness of knowledge graphs. Chapter 6 proposed a framework to generate reproducible link prediction experiments. This Chapter, follows on improving the evaluations of KGE methods by providing a better insight into the performance of KGE methods on individual relation patterns.

Available datasets for evaluation of link prediction methods do not consider different graph patterns, making it difficult to measure the performance of link prediction models in different knowledge graph settings. This drawback leads to leaking relation patterns in the current benchmarks, obscuring the understanding of the models' actual effectiveness on relation patterns.

This Chapter presents a diverse set of pragmatic datasets to facilitate flexible and problem-tailored Link Prediction and knowledge graph embedding research. We define knowledge graphs specified to different relational patterns in a diverse range of inductive characteristics, from being entirely inductive in one dataset to fully transductive test datasets. In addition, we generate various partially inductive datasets.

Using the extracted datasets, we embark on a comprehensive benchmark for the state-of-the-art KGE-based link prediction methods. We analyze the outputs of the methods on our datasets to compare their capabilities, provided that we consider uniform evaluation metrics for each dataset. Our analysis of datasets over link prediction models provides a better insight into the suitable parameters for each situation, optimizing the KG-embedding-based systems.

This Chapter targets the research question:

Research Question 4

How can we recognize a more effective embedding method on a specific relation pattern?

Contributions of this Chapter are as follows:

- Proposing a new benchmark for link prediction task that targets investigating KGE models on a single relation pattern basis.
- Proposing several datasets[1] by classifying triplets into their respective classes according to their patterns, keeping in mind the properties from both inductive and transductive types. Therefore, we extract four categories from each class: Fully Inductive, Fully Transductive, CountBased Inductive, and either Head or Tail Inductive. Each category is further divided into patterns of Symmetry, Anti-symmetry, Inverse, and Inductive, making of **32** datasets per category and **96** in total.
- Developing methods for extracting separated patterns and automatic rectifying methods to avoid data leakage between detests. The datasets are also designed based on unification to benchmark them onto different link predicting models.
- Observing a significant setback in benchmarking knowledge graph models, we extend the work done by [60], keeping in mind our set of data.
- Generating a fair comparison to help choose the best model and dataset combination, which is especially beneficial for NLP Research.
- The previous research on benchmarking datasets was too general; we provide a tool-set to designate a specific approach according to the type of datasets.
- Exploring the characteristics of the datasets that can be potential performance boosters.

This Chapter is based on following publication [7]:

- **Afshin Sadeghi**, Hirra Abdul Malik, Diego Collarana, Jens Lehmannn. *Relational Pattern Benchmarking on the Knowledge Graph Link Prediction Task*. Conference on Neural Information Processing Systems (NeurIPS 2021) 2021.

Some of the most used knowledge graphs include *DBpedia* [109], *Yago* [110], *Freebase* [111] and *WordNet* [85]. Despite being in demand, KGs still face many issues, such as data incompleteness. To tackle the issue, knowledge graph embedding models take the role of link predictors in which they observe the patterns in knowledge graphs based on how facts are connected together. According to [112], the goal of the link prediction task is to map the entities/relations to low dimensional vectors capturing the structure of the knowledge graph, which helps predict the likelihood score of the triple. Despite advancements in benchmarks, a significant chunk is still unexplored.

In this study, we enhanced the work of [113] by building various datasets on the principles of known and valuable facts using the Freebase and Wordnet datasets, categorizing them into different patterns for benchmarking.

[1] All datasets, scripts, and extended results are available: `https://github.com/mlwin-de/relational_pattern_benchmarking`

The relation of each triplet is observed and then grouped into categories. Each category includes relations involved in a specific relation pattern, e.g., symmetry has the "same" relation between two entities such as "friends", whereas inverse has two "different and directed" relations between the entities such as "father and son".We make categories of datasets and observe the link prediction (LP) models over them.

With the goal of setting up a benchmark that separates the task of testing KG embedding models from the models, we extend Benchembed, the MLwin-Hobbit platform for benchmarking various trained methods, which are implemented in different environments (e.g., PyTorch and Java). This extension is crucial for a reproducible evaluation and fair comparison of methods.

7.1 Integration into BenchEmbedd Benchmarking Platform

To perform a link prediction benchmarking based on FAIR principles, we integrated the models in BenchEmbedd [107], a platform that aims at benchmarking big linked data. Such benchmark experiments are frozen into docker containers, which can be accessed, reproduced, and reused easily with little prior knowledge of the test platform. The system allows researchers to make and test systems without having to worry about standardized hardware.

We train state-of-the-art KG embedding models from scratch with our RAW datasets, providing experimental results that give exciting research directions. We consider the most popular and unified evaluation metrics along with the AUC-PR test in all combinations. Our experiments suggest that the link predicting models are scalable to large-scale datasets and graphs. These results indicate fruitful guidance for future research in KG Link Prediction and KG Embeddings.

In the following Section, we review the related work on which we based our benchmarking and dataset generation. We highlight the different aspects of Relational datasets in the two Sections after it. They include relational patterns and different types of inductive and transductive settings for evaluation datasets. In Section 7.5, we describe our relation pattern-based extracted benchmark datasets, and our extraction method. Section 7.6 illustrates our Benchmark results which include an extensive set of state-of-the-art methods and our numerous relation pattern-specific datasets. In this Section, we particularly compare the two embedding methods proposed in Chapters 4 and 5 detailedly, and we finally conclude this Chapter in the Section 7.7.

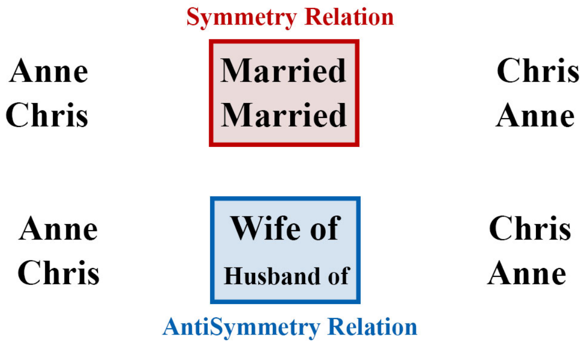


Figure 7.1: Example of triplet categorization based on relation pattern.

7.2 Related Work

Benchmarking datasets helps to compare and evaluate the LP models, parameters, and procedures as well as the statistics of the different datasets that are evaluated. The benchmarked dataset as described by [58] is useful for two basic analyses, efficiency, and effectiveness analysis.

7.2.1 Benchmarking

The dataset of [60] is kept as our ultimate guidance and standard to support the benchmarking task. CODEX dataset was benchmarked by [114] with unified evaluation strategies and empirical analysis. We studied and prepared subsets for Relational Patterns of Inversion, Symmetry, and Composition accordingly. We use the same strategy of making sub-datasets and then benchmarking on a number of models. Evaluation techniques of using MRR and Hit Ratios were considered but along with the introduction of **AUC-PR**. Using GNN and Graph Kernel methods as used by [115] gave us a new direction to use [51] GraIL for our dataset benchmarking. In the study of relation patterns, we include the most frequent patterns. In the related works, they are studies that consider experiments on more complex logical rules, such as [116] that evaluate Inverse Equivalence and Subsumption rules, and in this direction, [104] evaluates the performance of the knowledge base inference methods on a dataset of grid paths of different lengths.

The following Section dedicates to dataset building. We explain the importance of dataset building and refer to the current strategies that we based our study upon to make our benchmark datasets.

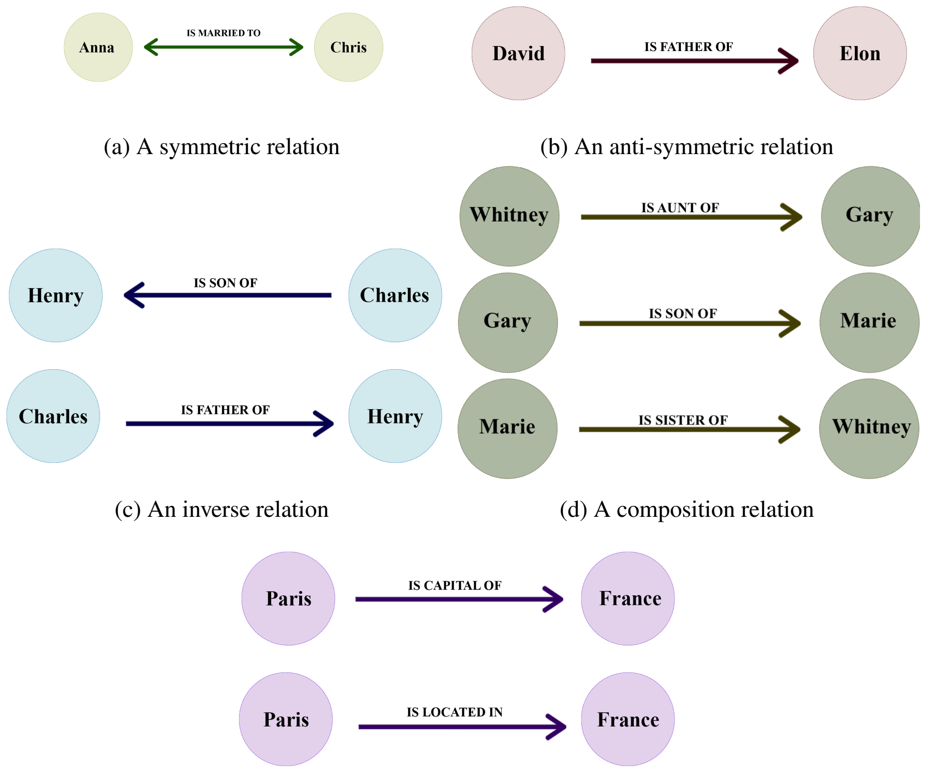
7.2.2 Dataset Building

Relation Extraction, the sub-field of information extraction, is one of the core techniques that support ML research. It organizes the structural information into groups according to the need [117].

We build our datasets keeping in mind the dataset building strategies from OGB [60] as well as the CODEX [114] and TU-Dataset [115]. CODEX is gathered from thirteen domains ranging from medicine to music. This dataset is built on the principles of snowball sampling to extract data, while in our study, we searched for specific relational patterns to enable pattern-specific evaluations. To have a fair comparison, CODEX uses a fixed set of negative/corrupted samples. We similarly use this approach for calculating AUC-PR scores in our evaluation. TU-Dataset, a unified set of over 120 datasets from several domains, targets graph classification and regression tasks, while in our study, we focus on the link prediction task.

We use the characteristics visualization technique for the datasets and the required properties for the analysis of the characteristics, of the aforementioned studies, as a baseline to define our dataset. We based our study on the two following datasets and extracted subsequent sub-datasets in the form of the stated patterns.

- **FB15k** A freebase dataset with a total of 592,213 triplets with 14,951 entities and 1,345 relationships. This factual dataset contains 483,142 Train triplets, 59,071 Test triplets and 50,000 Valid triplets. It dataset contains many entities from the wiki-link data.
- **WN18** A dataset extracted from Wordnet version 3 with a total of 141,442 triplets with 40,943 entities and 18 relationships. The dataset contains 141,442 Train triplets, 5,000 Test triplets, and 5,000 Valid triplets. The dataset supports text analysis and provides with dictionary/thesaurus. Lexical relationships between synsets are stated by this dataset.



(a) A symmetric relation

(b) An anti-symmetric relation

(c) An inverse relation

(d) A composition relation

(e) An inference relation

Figure 7.2: We define five different relational patterns to create our datasets and benchmark.

7.3 Pattern extraction from Relational Datasets

Pattern extraction is the core task in building datasets for machine learning research. Pattern type suggests the type of link prediction model that works best for the given dataset. Patterns, that are also expressed as rules, each have different suitability to the embedding models. Link Predictors learn the specific pattern of the datasets and then match rule-based patterns to provide reasoning. The patterns we considered to build our datasets from Relational Datasets are stated below:

7.3.1 Symmetry

This relational pattern is a sub-category of **Equivalence** pattern. Therefore, it is a binary relation that works in both directions. The relation can also be stated by the **Equal to** property, for instance, if a=b then b=a A relation is symmetric if:

$$\forall a, b \in X(aRb \Leftrightarrow bRa)$$

$$r(a, b) \Rightarrow r(b, a)$$

If R^T represents the converse of R, then R is symmetric if and only if $R = R^T$. Marriage, Friendship, and Partners are a few examples of symmetric relations.

7.3.2 Anti-Symmetry

The rule of Anti-Symmetry is opposite to Symmetry. It is a directed rule that states if a relation *R* binds A to B, the same can not work in the opposite direction, binding B to A. The rule is written as

$$r(a,b) \Rightarrow \neg r(b,a)$$

Relations such as Owner (to Tenant), Parent (to Child), and Singer (to Song) are Anti-Symmetric. Figure 7.2b shows a few examples of this rule.

7.3.3 Inverse

It is a binary relation stating two opposite relations for a set of entities. It is possible to assume a unique inverse relation for every relation. Inverse pattern between two set of two triples occurs when they have the opposite relation directions and have the same entities.

$$r_2(x,y) \Rightarrow r_1(y,x)$$

Parent-Child and Teacher-Student are examples of the inverse relations. Figure 7.2c shows how inverse relation is represented.

7.3.4 Composition

This binary relation which is also termed as **relation multiplication** is basically a **compound** relation which states the relation that can not exist without the existence of another relation. For Example, for the relation *Aunt*, the relation of *Sister* and *Son/Daughter* must exist in order to prove someone as an Aunt of somebody. In mathematical terms, relation r_1 is composed of relation r_2 and relation r_3 if:

$$\forall x, y, z : r_2(x,y) \wedge r_3(y,z) \Rightarrow r_1(x,z)$$

Figure 7.2d states that Paris is the Capital of France, According to the composition property, Paris must be in France to be its capital.

7.3.5 Inference

Inference relation pattern is one of the **logical** rules we formed datasets of. The rule states that we can deduce a relationship between two entities from the knowledge of another relationship between the two of them:

$$\forall h, t : (h, r_1, t) \Rightarrow (h, r_2, t)$$

where h and t are entities and r_1 and r_2 are relations between them. Figure 7.2e gives an example of inference relation which states Paris is the capital of France and thus, according to r1, Paris much be in France as well. Therefore, *(Paris, isCapitalof, France)* ⇒*(Paris, isLocatedin, France)*

Algorithm 2 Algorithm to extract dataset with a relation pattern P

```
Input: Knowledge graph K, relation pattern to extract P
Output: train$_{new}$, test$_{new}$ and valid$_{new}$ sets.
Generate dataset A = Union of all the triples $\in U(train, test, valid)$ of K.
for each triple t in A: do
    if found triples that form pattern P with t in A then
        Add the Premise triples forming the pattern to the set $S_p$
        Add the Conclusion triples forming the pattern to the set $S_c$
Add $S_p$ to the train$_{new}$ set.
for each relation pattern P' other than P do
    for each triple t in A: do
        if if found triples that form pattern P' with t in A then
            Add the Conclusion triples forming the pattern to the set $S'_c$
for For each triple $t \in S_c$ do
    if  t exist in $S'_c$  then
        remove it from $S_c$
Split $S_c$ into 3 parts $S_1$, $S_2$ and $S_3$
Add $S_1$ to train$_{new}$, $S_2$ to test$_{new}$ and $S_3$ to valid$_{new}$ sets.
Return: train$_{new}$, test$_{new}$ and valid$_{new}$ sets.
```

7.4 Evaluation Dataset Settings

There exist two different methods for the dataset division into train and test/validation subsets, where the composition of entities of each setting defines the evaluation setting:

Inductive Setting In an inductive setting, the entities during training are not found in the test dataset. The part of entities are kept missing and their relations are made to be found by the LP models. The number of disjoint entities varies in an inductively set dataset, fully inductive sets have fully disjoint set of entities and thus difficult for the models to predict. In our study we consider different ratios of disjoint entities and also different distributions of disjoint entities in the head and tail of triples.

Transductive Setting A dataset is divided to train and test/validation in a transductive setting when the occurrence of entities is ensured to be in the training procedure if it appears either in a test or valid sets. Transductive datasets are best for Entity Specific Embedding. All entities in the training set are present in the test set and thus a model has trained embeddings for them specifically.

7.5 A New Set of Pattern Specific Datasets

This Section describes our method to generate a new set of pattern-specific datasets for the link prediction task, followed by the setup description for benchmarking the state-of-the-are KGE methods using our extracted datasets.

Table 7.1: Inductive Setting Datasets

Inductive		**Train**	**Test**	**Valid**	**Total Entities**	**Total Relations**	**% of Original Dataset**
FB15k	**Symmetry**	4254	542	542	3447	51	0.901
	Anti-Symmetry	12930	3494	3884	8304	433	3.429
	Inverse	4753	2568	2568	7745	641	1.670
	Inference	3489	2824	2745	6083	611	1.530
WN18	**Symmetry**	2322	272	272	4344	5	1.893
	AntiSymmetry	16650	4698	4697	20552	18	17.203
	Inverse	8728	903	904	13842	17	6.958
	Inference	844	99	99	1639	15	0.688

7.5.1 The Standard Patterns

To make relation-specific datasets, we generated entirely disjoint sets from both FB15k and WN18 and then subcategorized them into relational datasets, creating a set of eight datasets. We extracted triples with a specific relation pattern to make each dataset. We then disregard common triples between two sets of relation pattern triples. Figure 7.1 shows an example of triple categorization based on relation patterns.

We developed Algorithm 2 that details this workflow. To extract a dataset for a relation pattern P from a benchmark knowledge graph K[2], it first unions the triples in train and test and validation sets. Then it extracts Premise and Conclusion triples that form a targeted relation pattern. It adds the Premise triples to the new train set. Then it searches the Conclusion triples for triples that leak other relation patterns into the extracted triples and excludes them. Finally, it divides the leftover Conclusion triples into three sets and adds these triples to the new train and test and validation sets. Based on this method, we extracted a subset of data from the standard FB15k and WN18 with Symmetry, Inverse, Anti-symmetry, and Inference patterns.

After generating datasets for each of the relation pattern, we extracted an entirely inductive and a transductive dataset and two more customized datasets. In the first set, the percentage of inductive and transductive triples is fixed, and in the second set, each triplet has one Inductive entity with the other entity being transductive. The description of these settings is in the following.

Inductive We built four datasets with the inductive setting where the entities of test and train datasets are entirely disjoint. Table 7.1 states the statistics of our datasets.

Transductive Transductive Setting has common entities in train and test datasets. Therefore, the entities are already seen by the model, making prediction much easier for them. Table 7.2 states the statistics of the set of eight datasets from the transductive type.

Head-Tail Ratio Inductive We built a set of datasets by keeping either the head or the tail of each triplet in the train hidden from the test dataset. By doing so, we gain a semi-inductive dataset with each triplet unseen. Table 7.3 reports the statistics of these datasets.

[2] The script to extract data based on each individual relational pattern is available in the code section of https://github.com/mlwin-de/relational_pattern_benchmarking/

Table 7.2: Transductive Setting Datasets

Transductive		Train	Test	Valid	Total Entities	Total Relations	% of Original Dataset
FB15k	Symmetry	5781	1399	1416	2823	52	1.452
	Anti-Symmetry	20711	128	109	2471	143	3.537
	Inverse	31332	750	750	10988	696	5.544
	Inference	70226	104	111	10500	377	11.895
WN18	Symmetry	1449	362	363	2030	5	1.436
	AntiSymmetry	6366	190	168	3393	15	4.441
	Inverse	4364	750	750	5765	17	3.873
	Inference	2027	12	10	3009	18	1.353

Table 7.3: Head-Tail Inductive Setting Datasets

Head-Tail Inductive		Train	Test	Valid	Total Entities	Total Relations	% of Original Dataset
FB15k	Symmetry	5621	989	990	3632	52	1.283
	Anti-Symmetry	15404	10795	10795	9241	470	6.247
	Inverse	24176	4701	4701	12065	794	5.670
	Inference	13845	5898	5636	9671	665	4.286
WN18	Symmetry	1630	185	186	2447	5	1.322
	AntiSymmetry	30000	5603	5603	23786	18	27.217
	Inverse	5421	621	621	7843	17	4.401
	Inference	462	84	85	815	15	0.417

Table 7.4: 50% Inductive Setting Datasets

Percentage Based Inductive		Train	Test	Valid	Total Entities	Total Relations	% of Original Dataset
FB15k	Symmetry	4677	445	444	3219	51	0.940
	Anti-Symmetry	14904	11603	11608	9911	472	6.436
	Inverse	6124	779	779	4958	600	1.297
	Inference	5840	5249	5263	7031	446	2.761
WN18	Symmetry	2009	253	253	3285	5	1.661
	AntiSymmetry	22208	5330	5329	21666	18	21.709
	Inverse	7613	678	678	10785	17	5.924
	Inference	685	58	64	1235	15	0.533

Percentage-wise building In our study, we generated half of the test triples with inductive settings and half with the transductive setting. We apply this percentage base data generation to each category. Table 7.4 describes the statistics of the 50% datasets.

For our benchmark, we take standard evaluation metrics of Hit Ratios (at 1, 3, 10), Mean Reciprocal Rank, Area Under the Curve, and AUC-PR. [118] suggests that for a perfect AUC-PR score, an equal number of negative triplets are needed along with positive triplets. Therefore, as described by [51], in the test set, the same number of negative samples are created by corrupting the copy of each triplet by either replacing the head or the tail with any random entity. We used the same procedure to incorporate each model with the AUC-PR score in a unified way.
We considered **DistMult** [12], **RotatE** [18], **TransE** [14], **GraIL** [51], **MDE** [15] and **CompGCN** [44] for our analysis.

Table 7.5: Hit@10 and MRR results of Link Predictors on datasets extracted from FB15k.

Type of dataset	Dataset	FB15k													
		DistMult		TransE		RotatE		MDE		GraIL		CompGCN		QuatE	
		Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
Inductive	Symm	0.0000	0.0003	0.0000	0.0002	0.0000	0.0002	0.0000	0.0002	0.0000	0.0228	0.0074	0.0060	0.0000	0.0040
	Anti-Sym	0.0000	0.0004	0.0000	0.0002	0.0000	0.0001	0.0000	0.0001	0.0000	0.0236	0.0023	0.0022	0.2435	0.0912
	Inverse	0.0021	0.0024	0.0028	0.0030	0.0019	0.0021	0.0019	0.0021	0.0000	0.0210	0.0012	0.0013	0.2530	0.0914
	Inference	0.0008	0.0008	0.0018	0.0020	0.0004	0.0005	0.0004	0.0005	0.0000	0.0235	0.0011	0.0013	0.2640	0.0982
Transductive	Symm	0.8755	0.8692	0.1405	0.0449	0.8594	0.8498	0.2184	0.1232	1.0000	0.9801	0.9836	0.8924	0.8604	0.7914
	Anti-Sym	0.0041	0.0031	0.0432	0.0155	0.0083	0.0041	0.5078	0.3129	0.9922	0.9836	0.9648	0.8540	0.2214	0.1780
	Inverse	0.0137	0.0087	0.0439	0.0160	0.0083	0.0069	0.1547	0.0881	0.9953	0.9307	0.8020	0.6305	0.2704	0.1240
	Inference	0.0000	0.0036	0.0800	0.0276	0.0100	0.0066	0.1827	0.1413	0.9932	0.9616	0.7308	0.5264	0.1309	0.0999
Head/Tail Ratio	Symm	0.0232	0.0097	0.0157	0.0061	0.0071	0.0043	0.0071	0.0047	0.4317	0.4065	0.0137	0.0068	0.0000	0.0166
	Anti-Sym	0.0019	0.0011	0.0009	0.0008	0.0008	0.0010	0.0084	0.0038	0.1226	0.1143	0.0066	0.0026	0.3997	0.1957
	Inverse	0.0031	0.0022	0.0341	0.0135	0.0059	0.0039	0.0849	0.0492	0.1380	0.1008	0.1272	0.0734	0.3393	0.1651
	Inference	0.0047	0.0032	0.0141	0.0054	0.0085	0.0046	0.0281	0.0132	0.0882	0.0709	0.0936	0.0583	0.3464	0.1785
Percentage Based(50%)	Symm	0.2596	0.2590	0.0079	0.0033	0.0461	0.0215	0.0674	0.0415	0.2584	0.2843	0.2629	0.2624	0.7500	0.5461
	Anti-Sym	0.0002	0.0005	0.0001	0.0003	0.0000	0.0002	0.0006	0.0006	0.0466	0.0422	0.0015	0.0013	0.3738	0.2146
	Inverse	0.0635	0.0534	0.0161	0.0076	0.0045	0.0028	0.0507	0.0367	0.1343	0.1554	0.1496	0.1310	0.2963	0.2121
	Inference	0.0098	0.0067	0.0142	0.0055	0.0217	0.0110	0.1369	0.1014	0.0782	0.0894	0.1484	0.1291	0.1966	0.1110

Experiment Setup Our system is implemented in Python, with Adadelta [23] as the optimizer. All Transductive bias models are set with learning rate **0.0001** with GraIL and CompGCN at **0.01** in-order to practice uniformity. Alpha (α) is kept between *[0.5,1]*. The models are set with the Dimensions = *[GraIL = 1000, CompGCN = 100 and all other models = 500]*. The epochs are set uniform for TransE, Distmult and RotatE = **6000** whereas MDE is given a higher number of **150,000** and GraIL and CompGCN are run are **100** and **500** epochs respectively. For TransE, Distmult, RotatE, and MDE, we used a fixed number of negative samples, 50, in all the experiments. We estimated the score for five runs and took the average to regulate loss function. All the experiments are performed on a local server with Intel Corporation Xeon E7 v4/Xeo CPU with 24 cores, 256 GB RAM, and GeForce GTX 1180 with 4 GPU cores.

7.6 Results

In this Section, we report the experiment results and discuss them. Tables 7.5 and 7.6 show the MRR and Hit@10 performance of the LP methods and the Tables 7.7 and 7.8 report the AUC-PR results. To provide a better overview of the AUC-PR results, Table 7.9, shows the average results of these two Tables in one place. Table 7.10 compares the results of MDE and the novel GFA-NN method, and Table 7.11 reports the ranking results of the LP models on the aggregate transductive datasets. Extended result sheets are available in `https://github.com/mlwin-de/relational_pattern_benchmarking`.

Inductive As far as Inductive datasets are concerned, in both FB15k and WN18, GraIL and CompGCN outperform all other models due to their property of inductive bias-ness. Segregating far, all models performed well over the Inference dataset. CompGCN gave a better performance on the Inverse dataset with a 56.46 AUC-PR score on the FB15k extracted dataset and 67.40 AUC-PR scores on the WN18 extracted dataset, whereas MDE did not perform well in all datasets from the inductive category. To sum up, the difference between Inductive and Transductive models could easily be noticed in this set of experiments.

Table 7.6: Hit@10 and MRR results of Link Predictors on datasets extracted from WN18.

Type of dataset	Dataset	WN18													
		DistMult		TransE		RotatE		MDE		GraIL		CompGCN		QuatE	
		Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR	Hit@10	MRR
Inductive	Symm	0.0000	0.0001	0.0000	0.0002	0.0000	0.0001	0.0000	0.0001	0.0000	0.0201	0.0018	0.0011	0.0000	0.0000
	Anti-Sym	0.0000	0.0002	0.0000	0.0001	0.0000	0.0001	0.0000	0.0001	0.0001	0.0304	0.0006	0.0007	0.0000	0.0016
	Inverse	0.0000	0.0001	0.0000	0.0001	0.0000	0.0001	0.0000	0.0001	0.0000	0.0215	0.0000	0.0003	0.0000	0.0009
	Inference	0.0000	0.0002	0.0000	0.0002	0.0000	0.0003	0.0000	0.0004	0.0000	0.0251	0.0051	0.0042	0.0000	0.0002
Transductive	Symm	0.9277	0.9064	0.0073	0.0031	0.7509	0.9792	0.4378	0.4195	1.0000	0.9979	0.9848	0.9738	0.9143	0.9466
	Anti-Sym	0.0000	0.0005	0.0000	0.0016	0.0000	0.0006	0.1079	0.0624	0.9684	0.9581	0.9816	0.9720	0.0374	0.0153
	Inverse	0.0000	0.0003	0.0000	0.0020	0.0047	0.0030	0.2220	0.1787	1.0000	0.9977	0.9940	0.9441	0.0377	0.0156
	Inference	0.2917	0.2394	0.0000	0.0013	0.0833	0.0523	0.2500	0.2520	1.0000	1.0000	0.2083	0.1566	0.0486	0.0323
Head/Tail Ratio	Symm	0.0027	0.0008	0.0000	0.0002	0.0055	0.0013	0.0162	0.0111	0.0054	0.0253	0.0081	0.0063	0.0272	0.0089
	Anti-Sym	0.0007	0.0005	0.0000	0.0005	0.0014	0.0014	0.0003	0.0003	0.0011	0.0291	0.0012	0.0006	0.0844	0.0453
	Inverse	0.0009	0.0007	0.0000	0.0004	0.0016	0.0015	0.0523	0.0318	0.0000	0.0340	0.0395	0.0196	0.0688	0.0577
	Inference	0.0000	0.0020	0.0000	0.0002	0.0000	0.0002	0.0833	0.0277	0.0000	0.0200	0.0714	0.0321	0.0198	0.0245
Percentage Based(50%)	Symm	0.5013	0.5029	0.0000	0.0003	0.0731	0.0437	0.0514	0.0353	0.5020	0.5119	0.0040	0.0030	0.8593	0.7899
	Anti-Sym	0.0000	0.0002	0.0002	0.0002	0.0000	0.0001	0.0005	0.0006	0.0111	0.0439	0.0010	0.0010	0.0198	0.0245
	Inverse	0.1924	0.1199	0.0118	0.0047	0.0701	0.0368	0.0155	0.0082	0.4808	0.4906	0.0000	0.0010	0.0193	0.0102
	Inference	0.0000	0.0002	0.0000	0.0005	0.0000	0.0007	0.0172	0.0174	0.0345	0.0552	0.0086	0.0121	0.1086	0.0196

Table 7.7 summarizes FB15k results and the green column of Table 7.9 summarizes the average AUC-PR on the Inductive setting. All models show a low performance on the MRR and Hit metric for the Inductive datasets. The results in the Tables 7.5 and 7.6 follow the same pattern. Nevertheless, the AUC-PR results show that some models can better distinguish positive and negative samples in an equal number of samples, e.g., CompGCN performs better in separating negative samples in the Inductive setting.

Transductive In evaluations with transductive setting datasets, all models showed significant improvement in the link prediction task than their performance over other datasets. TransE and MDE were almost 15% - 20% less accurate compared to other state-of-the-art models in the AUC-PR. The Symmetry dataset gives promising results with link prediction of more than 95% in almost all models. Furthermore, for the WN18 dataset, GraIL and even CompGCN predict 99% true triplets for the symmetry dataset. Overall, GraIL and CompGCN, and QuatE are proved to be the superior models for transductive datasets.

An exception of performance for the Symmetry dataset in Distmult between inductive and transductive cases was observed, where for both WN18 and FB15k the AUC-PR result increased in the transductive setting by an amount of about 40%.

Promising AUC-PR results for symmetry dataset by the model are displayed in Table 7.7 and Table 7.8.

Semi-Inductive - Head Tail Ratio Since this set of datasets is also inductive in its properties as all triplets are fully inductive with either one of the entities unseen. The models behave exactly the same way as they work with the inductive datasets. All state-of-the-art models fail to perform under such settings except GraIL and CompGCN due to their inductive nature. Despite the fact that GraIL and CompGCN perform better than the transductive bias model, they could not rank the triplets correctly with more than 0.52 AUC-PR. However, CompGCN shows some improvements for the Inference type datasets under the category with around 0.60 AUC-PR.

Table 7.7: AUC-PR Results of Link Predictors on datasets extracted from FB15k.

Type of dataset	DataSets	FB15k						
		Metric (AUC-PR)						
		DistMult	TransE	RotatE	MDE	GraIL	CompGCN	QuatE
Inductive	Symm	0.4933	0.4585	0.4554	0.4650	0.5000	0.7682	0.3115
	Anti-Symmetry	0.4993	0.4161	0.3889	0.4221	0.5009	0.6577	0.4841
	Inverse	0.4992	0.4632	0.4282	0.4496	0.4986	0.5646	0.4824
	Inference	0.5063	0.4578	0.4403	0.4527	0.5022	0.5566	0.4924
Transductive	Symm	0.9618	0.6901	0.9608	0.9434	0.9966	0.9995	1.0000
	Anti-Symmetry	0.5151	0.6038	0.5853	0.9986	0.9998	1.0000	0.8998
	Inverse	0.5345	0.6318	0.5967	0.7948	0.9963	0.9927	0.9137
	Inference	0.4907	0.7484	0.5359	0.7468	0.9971	0.9982	0.8933
Head/Tail Ratio	Symm	0.6409	0.7480	0.7259	0.7764	0.7846	0.6109	0.6526
	Anti-Symmetry	0.5054	0.5486	0.5406	0.5331	0.5881	0.5559	0.6539
	Inverse	0.5093	0.5507	0.5359	0.5499	0.5432	0.7815	0.6906
	Inference	0.5314	0.5401	0.5292	0.5156	0.5261	0.6410	0.6102
Percentage Based(50%)	Symm	0.7257	0.5955	0.6256	0.6201	0.6831	0.7591	0.7721
	Anti-Symmetry	0.4959	0.4474	0.4347	0.4208	0.5045	0.5774	0.7268
	Inverse	0.5870	0.5351	0.5602	0.5597	0.6206	0.7253	0.7368
	Inference	0.4935	0.5284	0.5140	0.5235	0.5620	0.6455	0.7169

Table 7.8: AUC-PR Results of Link Predictors on datasets extracted from WN18.

Type of Dataset	DataSets	WN18						
		Metric (AUC-PR)						
		DistMult	TransE	RotatE	MDE	GraIL	CompGCN	QuatE
Inductive	Symm	0.4912	0.4864	0.5164	0.4889	0.5000	0.4736	0.9892
	Anti-Symmetry	0.4972	0.4400	0.4119	0.4774	0.4999	0.6740	0.4616
	Inverse	0.5023	0.4312	0.3931	0.4667	0.5000	0.5248	0.4716
	Inference	0.5154	0.5175	0.4953	0.5131	0.5000	0.5398	0.4242
Transductive	Symm	0.9855	0.6506	0.9849	0.9395	1.0000	0.9993	1.0000
	Anti-Symmetry	0.5705	0.6048	0.5183	0.9856	0.9842	0.9953	0.9875
	Inverse	0.4507	0.5274	0.6299	0.9244	1.0000	0.9952	0.9683
	Inference	0.8745	0.7127	0.8655	1.0000	1.0000	0.7179	0.8913
Head/Tail Ratio	Symm	0.5390	0.5609	0.6477	0.6652	0.5053	0.6172	0.6449
	Anti-Symmetry	0.5328	0.5139	0.5135	0.4969	0.4999	0.5722	0.6258
	Inverse	0.5202	0.5746	0.5442	0.6060	0.5000	0.5812	0.7249
	Inference	0.5905	0.4181	0.6306	0.8024	0.5000	0.5145	0.6294
Percentage Based(50%)	Symm	0.8185	0.5920	0.7657	0.7487	0.7626	0.5888	0.7873
	Anti-Symmetry	0.4942	0.4773	0.4836	0.4769	0.5103	0.5973	0.8900
	Inverse	0.7936	0.6586	0.7142	0.6418	0.7974	0.5374	0.8134
	Inference	0.5472	0.5291	0.5290	0.5134	0.5031	0.5150	0.7924

Semi-Inductive - Percentage-Wise The datasets are gathered on the principle of half datasets from the inductive category with 50% triplets of the transductive nature. Due to this reason, they perform well on the link prediction models. Distmult and CompGCN give promising results of more than 85% on the symmetry datasets. TransE could not give a better score due to its inability to infer symmetric patterns, although it predicts inference patterns better than any other. As an average, we conclude that all models outperform over the transductive setting of datasets in the link prediction task. The KGE models also give promising results over datasets with the half inductive and half transductive type (Count-Based Dataset), evidently due to the fact of containing exactly 50% transductive triplets. Inductive datasets and head-tail inductive datasets still face issues in the link prediction task with better results on CompGCN and GraIL.

Table 7.9: Mean AUC-PR performance of the LP methods.

	WN18				FB15k			
Type of dataset	Metric (AUC-PR)							
	Inductive	Transductive	Head/Tail Ratio	Percentage Based(50%)	Inductive	Transductive	Head/Tail Ratio	Percentage Based(50%)
DistMult	0.5015	0.7203	0.5456	0.6634	0.4995	0.6255	0.5468	0.5755
TransE	0.4688	0.6239	0.5169	0.5643	0.4489	0.6685	0.5969	0.5266
RotatE	0.4542	0.7497	0.5840	0.6231	0.4282	0.6697	0.5829	0.5336
MDE	0.4865	0.9624	0.6426	0.5952	0.4474	0.8709	0.5937	0.5310
GraIL	0.5000	0.9960	0.5013	0.6433	0.5004	0.9975	0.6105	0.5926
CompGCN	0.5531	0.9269	0.5713	0.5596	0.6368	0.9976	0.6473	0.6768
QuatE	0.4426	0.6518	0.6518	0.7382	0.5867	0.9618	0.6563	0.8208

AUC-PR Performance Summary Table 7.9 summarises the previous reported AUC-PR results by averaging them. In this extract, we observe that the entirely inductive test is the most challenging for the models, and then the Head/Tail Ratio and the Percentage Based tests were among the challenging tests. The Transductive experiments were the least challenging of all the methods.

The models showed similar performance on the Inductive WN18 experiments with less than a 0.05 difference in the measure. The models produced results on a more diverse range in the Inductive FB15k tests. In contrast, the difference in the results in the Transductive tests was the most significant.

7.6.1 Discussion

Inductive Evaluations Since none of the three types of inductive experiments included known triples in the test phase, the models did not use any other information indicative of the identity of tested entities (e.g., their distance to other entities), except for GraIL which applies sub-graph calculations for the test-set as well. Thus, the vector weights of the unknown entities are set randomly. Consequently, we conclude that the experimented models that benefit from no knowledge of entities estimate the rank of the samples only by generating an embedding space such that the scores of negative triples are separable from the random values to some extent using the common relations between test and the training dataset.

Our second observation is that while the tested models perform with scores mostly above 0.40 on the AUC-PR test, they are less effective on the more difficult Hit and MRR ranking measures in the inductive evaluations. The number of negative samples is a fundamental disparity between the AUC-PR and MRR tests. In AUC-PR, their number is equal to the number of positive samples, and in Hit and MRR, the number of negative samples is equal to the number of all the entities of a KG. Therefore the larger a dataset is, the Hit and MRR test become more challenging, particularly in the Inductive setting. Therefore we suggest that KGE studies consider the effect of the size of the test knowledge graphs in evaluating their method.

In the following, we generally compare the performance of models over datasets extracted from the two original knowledge graphs.

FB15k For FB15k, GraIL and CompGCN perform well on the inductive datasets, with the highest performance on the Symmetry dataset in CompGCN, whereas GraIL shows a near performance on all datasets with the inductive setting.

Table 7.10: Evaluation results of GFA-NN model compared to MDE.

LP Methods	Type of DataSet	WN18				FB15k			
		Metric (AUC-PR)							
		Symm	Anti-Symmetry	Inverse	Inference	Symm	Anti-Symmetry	Inverse	Inference
MDE	Inductive	0.4889	0.4774	0.4667	0.5131	0.4650	0.4221	0.4496	0.4527
	Transductive	0.9395	0.9856	0.9244	1.0000	0.9434	0.9986	0.7948	0.7468
	Head/Tail Ratio	0.6652	0.4969	0.6060	0.8024	0.7764	0.5331	0.5499	0.5156
	Percentage Based(50%)	0.7487	0.4769	0.6418	0.5134	0.6201	0.4208	0.5597	0.5235
GFA-NN	Inductive	0.4572	0.4218	0.3861	0.4877	0.4593	0.3939	0.4477	0.4584
	Transductive	1.0000	0.9995	1.0000	1.0000	1.0000	0.9999	0.9848	0.9695
	Head/Tail Ratio	0.6653	0.4977	0.6409	0.5846	0.6289	0.5470	0.6425	0.5764
	Percentage Based(50%)	0.7600	0.4662	0.7301	0.5382	0.6842	0.4264	0.6024	0.5437

CompGCN also performs well on transductive datasets with almost 100% accuracy on Anti-Symmetry, along with outstanding performance on all datasets with the Count-based setting. QuatE and GraIL show the same trend as CompGCN with near accuracy.

MDE performs well on transductive setting datasets with 99% accuracy on the Anti-Symmetry followed by Head-Tail Ratio based datasets. RotatE, TransE and DistMult showed the same performance pattern as MDE, with the highest performance in the transductive setting and least in inductively set datasets. Table 7.7 describes the result of these evaluations.

WN18 In datasets extracted from WN18, a similar pattern as the FB15k dataset is observable. GraIL and CompGCN show better results on inductive settings than other models. Link prediction in the inductive and the transductive setting is better on WN18 datasets in most models as compared to FB15k, which evidently is due to fewer relations to be computed. Link prediction on Symmetry datasets from the transductive setting has around 0.99 AUC-PR for almost all the models considered. Table 7.8 quotes results on WN18 Datasets.

7.6.2 Evaluation of GFA-NN model

Finally, we benchmark our datasets over a newer multi-objective optimization KGE similar to MDE, i.e., GFA-NN [27], which considers the datasets' graphical features to create embeddings. The node features and graph properties are calculated and stored in separate files for training purposes. The graphical feature information is not available to the model in the evaluation phase to make a fair comparison.

Comparison between the previous MDE and GFA-NN reported in Table 7.10 shows outstanding behavior of the newly introduced method in the Transductive and Semi-inductive settings. The most significant improvement is observable in the Inference datasets, where the results show an improvement of between 0.6% to 22% in the tested cases. For Instance, the FB15k Inference dataset with the transductive setting improved from 74% in the AUC-PR score to 96%. Lastly, the results in the purely inductive setting only show an improvement in the Inference dataset. That is because we handicapped GFA-NN, not knowing the graphical features of any nodes in the test dataset.

Table 7.11: Ranking results of the LP models on the aggregate datasets.

	WN18RR			**FB15k-237**		
Model	MR	MRR	Hit@10	MR	MRR	Hit@10
QuatE	–	0.482	0.572	–	**0.366**	**0.556**
TransE	357	0.294	0.501	357	0.294	0.465
DistMult	5261	0.44	0.49	254	0.241	0.419
CompGCN	3533	0.479	0.546	197	0.355	0.535
RotatE	3340	0.476	0.571	177	0.338	0.533
MDE	3219	0.458	0.536	203	0.344	0.531
GFA-NN	3390	**0.486**	**0.575**	186	0.338	0.522

7.6.3 Comparison to Aggregate datasets

Table 7.11 shows the results of the LP methods on the aggregate datasets reported in [27] and [18].

This comparison of this table results to our pattern-specific experiments shows that while the difficulty of each individual relation pattern for each model influences the overall results, the ratio of each relation pattern also impacts their performance because the results are averaged in the MRR and Hit evaluations. For example, while CompGCN is not the best method for learning inverse relations in the transductive setting, it is one of the most effective methods in the FB15k-237 because FB15k-237 and WN18RR are missing the inverse relations from FB15k and WN18.

7.7 Conclusion

To sum up, we created several standard datasets with distinct relational patterns ranging from symmetry to inverse and different degrees of inductiveness. We evaluated several state-of-the-art Link Prediction models over them. We extended our research by working on link prediction models by this benchmarking approach and highlighting their work on different types of datasets. A unified evaluation strategy of AUC-PR measurement is incorporated into all link prediction models besides the Hit@K and MRR measures. We highlighted the datasets with improved performance on particular LP models. In order to support further research in the domain, we incorporated our benchmarking datasets to the BenchEmbedd for evaluation of a linked data life-cycle. The meaningful experiment outcome indicates that these datasets will foster the research on Link Prediction and KG embeddings.

CHAPTER 8

Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter

In previous Chapters, we proposed methods to improve knowledge graph embedding and more competently evaluate such methods for the link prediction task. Depending on the application context, the semantics of targeted hidden links in the link prediction task take on different meanings. In this Chapter, we define a novel notion and application for link prediction using knowledge graph embedding.

Informing professionals about the latest research results in their field is a crucial task in the field of health care because any development in this field directly improves the health status of the patients. Meanwhile, social media is an infrastructure that allows the public to share information instantly. Thus it has recently become popular in medical applications. In this study, we apply multiple distance knowledge graph embedding (MDE) model from Chapter 4 to link physicians and surgeons to the latest medical breakthroughs shared as the research results on Twitter. Our study shows that by using this method, physicians can stay informed about the new findings in their field, given that they have an account dedicated to their profession.

This Chapter targets the research question:

Research Question 5

Does a KGE method that allows the encoding of relation patterns tackle a real-world link prediction task more effectively than a commonly applied KGE?

Contributions of this Chapter can be summarized as follows:

- Proposing a pipeline for applying the MDE machine learning method as a recommender system to help physicians stay up-to-date in their field.
- Proposing a method for extracting and processing stream data from Twitter Social Network and generating a knowledge graph from the data.

- Proposing an extension of the multiple distance embedding method (MDE) that gives a probability of a predicted link based on the learn relations from the Twitter knowledge graph.
- Evaluating the link suggestion mechanism for connecting physicians to the latest medical study results meaningfully relevant to them.

This Chapter is based on the following publications [15, 119]:

- **Afshin Sadeghi**, Jens Lehmann. *Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter*. The 4th Workshop on Data Science for Social Good - SoGood 2019. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2019) 2019, 622–630.
- **Afshin Sadeghi**, Damien Graux, Hamed Shariat Yazdi, Jens Lehmann. *MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs*. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) 2020, 1427–1434.

Twitter is a projection of the interactions of a society connected to the internet, which is in constant evolution. The dynamic aspect of this social media allows manifold applications. With the rise of social media, Twitter was used to measure campaign impacts, collect opinions, analyze trends, and study crises. However, recently, its applications have been more individualized. Particularly, because Twitter has become the most popular form of social media used for healthcare communication [120], and it is reshaping health care [121], it has become the center of many studies in the field of health care. For example, a study suggests Twitter for knowledge exchange in academic medicine [122], and it was argued that disease-specific hashtags and the creation of Twitter medical communities [123] had improved the uniformity of medical discussions. Another study is dedicated to the influence of specific medical hashtags on social media platforms [124].

Problem Statement: Pershad et al. [125] point out the potential of Twitter to reshape public health efforts, including disseminating health updates and sharing information about diseases. Especially, they emphasize the role of Twitter in making research advances more accessible for physicians. They argue that connecting researchers and clinicians is crucial and valuable since clinicians can use new information they discover from this closer contact with researchers to guide decision-making about patient treatments in a field that is in constant progress.

In this study, we target this problem by providing a method that suggests to physicians and clinicians the recent research breakthroughs in their specialized field based on their current social activity. As the first step to reach this goal, we extract a subset of the Twitter network, and we generate a knowledge graph (KG) from the extracted data. Figure 8.1 depicts a schema of the KG with example user instances and the relations between them. In this figure, it is shown that our method recommends a Tweet of Jane, who is a researcher, about her latest findings to Bob who is a surgeon. The method calculates the probability that such a Tweet will be useful to Bob based on his previous favored Tweets and the relation to other physicians that work in the same field. We then apply an embedding method to predict links that are likely to be serving the physicians. The proposed application is different from the user recommendation service of Twitter [126] which recommends users to follow or the works that discover similar users [127]. Here, we focus our study and evaluation on suggesting related Tweets.

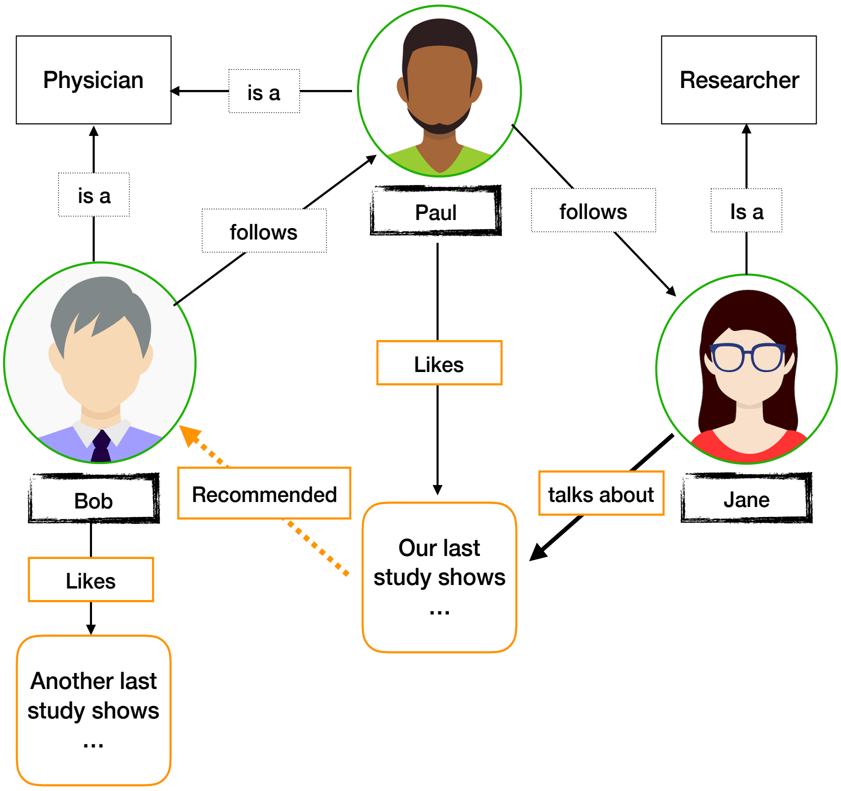


Figure 8.1: This diagram displays a schema of an example medical professional knowledge graph. This sample of the Twitter network contains three users and their relations. It depicts Jane as a researcher that publishes Tweets about her latest papers. Paul is a physician following her updates, and Bob is a third user that follows Paul and is a physician but does not know about Jane and her publications. The orange dot line depicts a new link suggested by our proposed method.

In the following Section, we give a brief background of this study and the definition of concepts used in this Chapter. Then in Section 8.2, we review machine learning methods that predict links on social networks and briefly explain a model that we use as the gold standard in our evaluations. In Section 8.3, we propose and explain a method to apply knowledge graph embedding on Twitter data to connect physicians to the latest research results in the medical field, and Section 8.4 presents our experiments, where we put our proposed approach under empirical tests. We continue with discussion of the experiment results in Section 8.5 and we conclude this Chapter in the Section 8.6.

8.1 Background

In a social network, a graph node represents a person, while edges that link the nodes correspond to relationships between people. The edges are also called “connections” or “links”. Examples of social networks are graphs that describe Facebook and Twitter. Link prediction, in general, is the task of predicting whether a link exists between a given pair of nodes or not.

Definition 1: a knowledge graph is defined by (E, P, T), a set of entities e ∈ E, a set of predicates p ∈ P, and a set of triples t ∈ T. A triple (e_i, p_k, e_j) is made of two entities and a predicate that connects them.

In a KG, two entities can be connected by several predicates. When describing a social network by a KG, nodes are translated to entities, and links are translated to predicates. However, in a KG, an ontology usually specifies the class that describes what types of entities and predicates can construct a triple. A relational learning model usually learns the relations of a KG. Embedding models are a class of relational learning models that produce vector representations of the entities and predicates and predict the missing links.

Definition 2: link prediction in a KG means to predict the existence of a triple, i.e., whether a relation exists from two entities e_i and e_j and a k-th predicate.

8.2 Related Work

Classic link prediction methods on social media use graph properties of the social network or NLP feature of nodes to predict links between entities. For example, [128] is based solely on graph features and [129] uses a similar technique for the social networks in healthcare. Meanwhile, [130] uses common words to cluster and rank nodes and, based on that, predicts the closely-ranked nodes to be connected. Another Study [131] uses a combination of graph features and keyword matches to train classifiers(SVM, Naive Bayes, etc.) to predict if a link exists between two nodes.

Most of the studies on link prediction of social networks focus on the problem of link existence. In contrast, some methods attempt to find link weights and the number of links between the nodes [132]. An advantage of KGE-based link prediction is that a KGE also predicts the link's type since KG embedding models distinguish the kind of links.

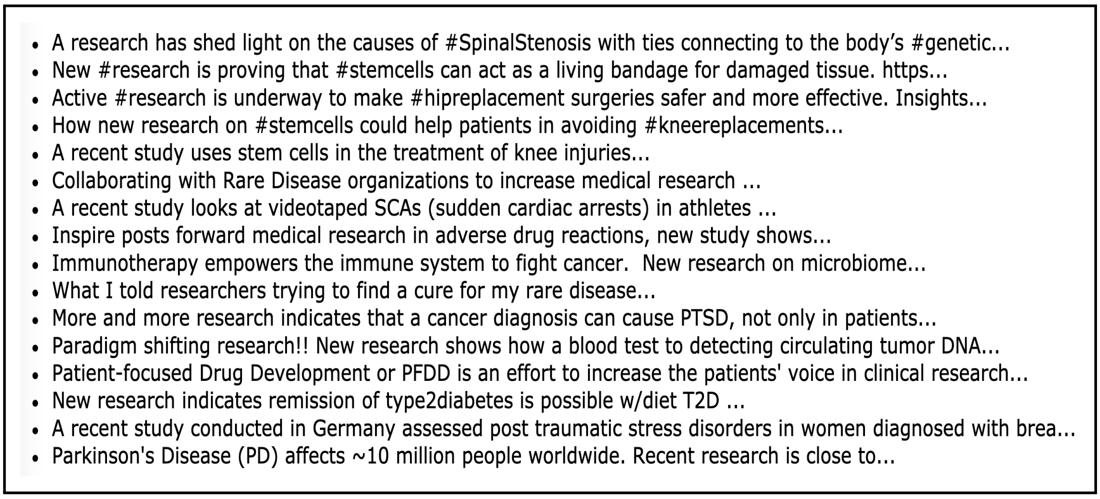
TransE [14] is an embedding model that is popular because of its simplicity and efficiency. It represents the entities in a KG by a relation between the vectors representing them. The score function describing these vectors in TransE is:

$$Score_{TransE} = \| h_i + r_i - t_i \|_p \qquad (8.1)$$

where n refers to L_1 or L_2 norm and h_i and t_i are the vector representations of an entity and r_i is the vector representations of a predicate. For training, TransE uses margin ranking loss as the loss function and scales well on the large datasets. Since this method is well studied in the literature, we consider it the gold standard method for our experiments. The following Section describes embedding a KG extracted from Twitter using the MDE model and TransE and describes a method to evaluate predicted links by the KGE models to estimate plausible links.

8.3 KG Embeddings for Twitter Link Prediction

Knowledge graph embedding models usually generate a prediction based on their score function. Nickel et al. [22] suggest performing link prediction by comparing the score of a triple with some given threshold θ or by ranking the entries according to their likelihood that the link in question exists.



- A research has shed light on the causes of #SpinalStenosis with ties connecting to the body's #genetic...
- New #research is proving that #stemcells can act as a living bandage for damaged tissue. https...
- Active #research is underway to make #hipreplacement surgeries safer and more effective. Insights...
- How new research on #stemcells could help patients in avoiding #kneereplacements...
- A recent study uses stem cells in the treatment of knee injuries...
- Collaborating with Rare Disease organizations to increase medical research ...
- A recent study looks at videotaped SCAs (sudden cardiac arrests) in athletes ...
- Inspire posts forward medical research in adverse drug reactions, new study shows...
- Immunotherapy empowers the immune system to fight cancer. New research on microbiome...
- What I told researchers trying to find a cure for my rare disease...
- More and more research indicates that a cancer diagnosis can cause PTSD, not only in patients...
- Paradigm shifting research!! New research shows how a blood test to detecting circulating tumor DNA...
- Patient-focused Drug Development or PFDD is an effort to increase the patients' voice in clinical research...
- New research indicates remission of type2diabetes is possible w/diet T2D ...
- A recent study conducted in Germany assessed post traumatic stress disorders in women diagnosed with brea...
- Parkinson's Disease (PD) affects ~10 million people worldwide. Recent research is close to...

Figure 8.2: A sample of the extracted Tweets about the recent medical studies. Each row shows the content of one of the extracted Tweets.

We similarly use the Multiple-Distance Embedding (MDE) model [15]. In comparison to TransE, this model can learn several relational patterns and thus it can more accurately learn the hidden relation between the entities. Specifically, MDE can learn relations with symmetry, antisymmetry, transitive, inversion, and composition patterns. The overall score function of this model is as follows:

$$Score_{MDE} = w_1 \parallel h_i + r_i - t_i \parallel_p \ + w_2 \parallel h_j + t_j - r_j \parallel_p \ + w_3 \parallel t_k + r_k - h_k \parallel_p - \psi \tag{8.2}$$

where $\psi \in \mathbb{R}^+$ is a positive constant. The loss function of this model is:

$$loss = \beta_1 \sum_{\tau \in \mathbb{T}^+} [f(\tau) - \gamma_1]_+ + \beta_2 \sum_{\tau' \in \mathbb{T}^-} [\gamma_2 - f(\tau')]_+ \tag{8.3}$$

where γ_1, γ_2 are small positive values and $\delta_0, \delta'_0 = 0$. $\beta_1, \beta_2 > 0$ are constraints. The given loss minimizes the score of the positive samples. Therefore, the smaller the triple score, the more probable the relation. Based on this property of the loss function, we define a measure to estimate the existence of a predicate such that the more probable triples are given a higher score.

We designate the division of the maximum score of a triple in the training set to the score of a triple as the probability of its existence:

$$P_a = \frac{max(Score_{training-triples})}{Score_A} \tag{8.4}$$

This definition is based upon the assumption that after the training, the model accurately predicates the triples of the training set. The equation compares only the triples of the same type (with the same predicate). Thus, in predicting the triples for linking physicians to medical Tweets, we consider only triples with $like\_research\_Tweet\_id$ predicates.

To perform link prediction on Twitter, we train MDE over an extracted KG. In the following, we explain the procedure to extract the dataset from which we later generate a KG using it.

Relation Id	Relation
0	is_talking_about
1	is_followed_by
2	is_following
3	job_title_type_is
4	likes_research_Tweet_id

Table 8.1: Relation types in the Social Ontology

Class Id	User Entity Class
0	job_title_medical_researcher
1	job_title_physician

Table 8.2: Class of users in the Social Ontology

Knowledge Graph Extraction: We extract a set of Tweets about the latest medical studies using Python scripting and the Tweepy library [1]. We filter our search by medical keywords and time in order to only obtain medical research-related Tweets which were created from the beginning of the year 2019. Figure 8.2 shows a sample of the extracted Tweets. To keep the privacy of users, we removed the user and Tweet identifiers from the figure.

We extend our approach to extract Twitter users who are physicians, surgeons, nurses, and researchers in the medical fields who have written about these topics or favored such Tweets. Our continuous inquiry, with a duration of 8 hours, provided 5996 Twitter users. Between these users, the job title of 69 instances was deductible (researcher in the medical field or physician) based on the medical job titles in their profile descriptions. We then grow a sub-graph using these 69 instances as stem entities and extract their user-user and the user-Tweets relations. The extracted user-user relations involve the users who follow or are followed by these users so that we generate the social neighbors of these users in the Twitter network.

In the next step, we define an ontology for the extracted interactions. Our created ontology includes five types of relations. Table 8.1 lists these relation types. We also anticipate two classes for users in the ontology. Table 8.2 presents these classes. We generate a multi-relational knowledge graph based on the ontology and the scraped data. This step converts our extracted relations to triples. The generated knowledge graph TW52 includes 4439 entities comprising 1021 users and 3418 Tweets. In this dataset, we only include these users' user-to-user relations and user-to-Tweet relations. The final constructed KG includes 4791 triples. The anonymized dataset is available for research in https://github.com/afshinsadeghi/SocialEmbeddings.

8.4 Experiments

We set up two experiments. We first evaluate how well the MDE method performs on the social media dataset against a baseline in the link prediction task. We then analyze the suggestion results of the model in different situations.

8.4.1 Performance Evaluation

We set up an experiment to evaluate the link prediction performance of MDE against TransE as the baseline.

[1] https://www.tweepy.org/

Evaluation Setup: We dedicate 80 percent of the knowledge graph extracted from Twitter as the training dataset and set the rest as the test dataset. We randomly choose triples by uniform random selection to separate them for the test set. We perform ranking the score of each test triple against its versions with replaced head and once with a replaced tail. We then compute the hit at N(hit@N), mean rank(MR), and mean reciprocal rank (MRR) of these rankings. We set the vector size of TransE to 20 and choose the vector size of 10 for MDE. We use L_2 normalization to normalize their score function and train them by 700 iterations. For MDE, we set the hyperparameters as follows: $\gamma_1 = \gamma_2 = 3$ and $\psi = 1.2$.

Results: Table 8.3 lists the evaluation results of TransE and MDE on the extracted knowledge graph. Due to the sparsity of the graph, TransE gains very low ranking scores while MDE produces superior results for all the MR and MRR and hit@N tests. The results suggest the positive influence of relation patterns learning in MDE.

Model	MR	MRR	Hit@1	Hit@3	Hit@10
TransE	1327	0.021	0.005	0.019	0.048
MDE	**1287**	**0.148**	**0.071**	**0.161**	**0.332**

Table 8.3: Results on Twitter extracted dataset (TW52). Better results are in bold.

8.4.2 Link Prediction Analysis:

In this Section, rather than studying the model's performance, we establish an experiment to analyze the suggestion results of the model to find out whether it creates sound suggestions in different situations. We apply the model to encode the constructed KG, and then we use it to suggest the possibly engaging research results for the physicians. We then study the recommended results.

Considering the physicians in the KG and Tweets, which include research results, we calculate the probability that such a Tweet is favorable for physicians using Equation 8.4. In our experiment, the hit@1 of the training triples was 99.8 percent. Therefore, assuming maximum probability for the training triples in the formula holds for the experiment.

The observation of relations and entities in the KG shows that it is structured with the small world network patterns [133]. Particularly, it includes hub users and Tweets which are connected to other nodes with a number of links that greatly exceed the average degree in the network.

We select a subset of physicians in the KG and classify them according to their relation to other users and Tweets into four groups of 5 users. We particularly inspected their relation to hub users, which we call User type A. Users of type A are followed by a large number of users(at least 200), they are active users and have favored variant Tweets. We also consider users of type B who follow a small number (25) who are also physicians or researchers. Table 8.4 lists these groups of users and the mean of their probability of like a Tweet C that includes a research study. We consider two Tweets similar if their representative vectors have a small angle. These Tweets are usually favored by the same group of people.

It is observable from Table 8.4 that in the proposed model, users that follow a diverse group of users and topics are less likely to be interested in an inquired Tweet than those with less diverse connections.

User group	Mean Probability of C
Users U that follow A. A and U like a Tweet similar to C	0.205
Users U that follow A. A likes a Tweet similar to C	0.134
Users U that follow B. A and B like a Tweet similar to C	0.975
New users U that still follow nobody and like no Tweet	0.127

Table 8.4: Mean Probability of linking to a Tweet C for users with different communities and liked Tweets

This effect is even stronger than when the user has liked a similar Tweet before. This suggests that the model performs better if a Twitter account is dedicated only to social communications related to her profession. Additionally, the new users that have not favored any Tweet are expected the least among the users to favor a Tweet.

8.5 Discussion of the specificity of the problem

The proposed experiment in the study has two major components. The first is the data extraction and KG construction part, in which we specify the problem by data cleaning and filtering the extracted data and create an ontology specific to the physicians and research-related tweets. The result of this part is TW52 knowledge graph which has a sparse network structure compared to the conventional benchmark datasets of embedding models, i.e., WordNet18 and FB15k.

The second part of the study is the MDE embedding model. Although MDE is a general method for the link prediction problems, the evaluations showed that it is capable of embedding the sparse dataset much better than the state-of-the-art TransE model. Therefore we consider both components appropriate for the proposed problem.

8.6 Conclusion

This Chapter proposed the application of multiple-distance knowledge graph embedding (MDE) to suggest Tweets about medical breakthroughs to physicians. We extracted a KG of medical research Tweets and their relations to the users which are medical researchers, or physicians.

We evaluated MDE against TransE as the baseline in a link prediction test for the social network KG. Our experiment shows the superior ranking performance of MDE over the baseline. We defined a probability for link suggestion and provided an analytic study. We thereby conclude that the model can be suggested to serve in connecting the physicians and the up-to-date advances in medical studies. Considering physicians' time constraints on social media [134], automating such suggestions can help physicians find news and trends relevant to medical research results more effectively and in less time.

For future work, it would be interesting to extend this study on a large scale and provide it as a live service. In addition, future studies may investigate the social effect of such application to find its effect benefits for patients besides the physicians.

CHAPTER 9

Conclusion and Future Directions

In this thesis, we follow the research objective of improving the state of the art of KGE for complex knowledge graphs. Chapter 1 identifies the research problem and discusses the key challenges to address in order to reach the research objective.

We review the fundamentals and background concepts essential for this thesis in Chapter 2. Chapter 3 covers the related work corresponding to the research objective and research question. In Chapter 4, we develop a novel learning method for knowledge graph embedding that removes a KG embedding capability limit, i.e., being limited to one score function. We propose this idea as the basis of this study and materialize the vision by defining independent vectors for a multiple distance embedding method with several score functions. We present several score functions that allow the learning of several relation patterns.

We further extend the idea of multiple distance objectives in learning global node features in Chapter 5. We show the limitations and capabilities of this approach in theory and empirically. Furthermore, we demonstrate the greater efficiency of the model in comparison to the state-of-the-art approaches. We observed a more pronounced improvement in the inductive setting and on a large-scale biological dataset. Chapter 6 dedicates to our effort in facilitating research on embedding methods, where we develop a benchmark generating framework that generates reproducible evaluation experiment units.

This study in Chapter 7 targets another complexity in evaluating relational pattern learning. Because the link prediction benchmark datasets are mixing relation patterns, it is unclear how a method's efficiency is on an individual relation pattern. Therefore, we create a group of leak-free datasets for four frequent relation patterns in the Inductive and Transductive settings. We conduct a benchmark based on these datasets that spot the difference in the efficiency of several state-of-the-art embedding methods in different situations. The results of this study can be used for model selection based on the characteristics of target datasets.

Last but not least, Chapter 8 provides a proof of concept demonstrating the application of KGE methods in the digital medicine branch. We illustrate how our proposed MDE method works in the deployment step of an End-to-End machine learning workflow to help physicians stay up-to-date in their work domain.

9.1 Review of the Contributions

This Section concludes the thesis by reviewing our research questions and their contributions, now taking into account the principal achieved findings that validate our research questions.

Research Question 1 (RQ1)

Does combining multiple distance-based scores targeting different relation patterns generate more effective embeddings of knowledge graphs?

The capability to allow the modeling of different relation patterns plays a crucial role in embedding knowledge graphs. In Chapters 4 we addressed **RQ1** by exploring an aggregate loss optimization approach for multiple distance objectives. We developed this solution based on the idea of independent learning vectors coupled with a limit-based loss function. The output of this study is the MDE knowledge graph embedding method that allows the encoding of several relation patterns. We formulated each score in this aggregate optimization method such that each one targets one challenge in learning one of the relation patterns. Here, to better understand how to determine each score correctly, we analyzed different scores from the state-of-the-art models and showed why they fail in learning particular relation patterns.

The involved independent vectors per score in this method do not increase the memory consumption of the approach out of the linear complexity and yet allow combining contrasting modelings of triples. The limit-based loss separates the modeling of positive and negative triples, allowing the method to learn them independently. In addition, this technique facilitates the analysis of the combination of the scores. Furthermore, we developed an algorithm that explores the space of scores for negative and positive samples and the generated overall loss to find an optimized limit for the limit-based loss during the training iterations. We conducted experiments that empirically demonstrated how well the involved scores were learning their targeted relation patterns and performed ablation studies that show how adding/removing each score function improves/drops the efficiency of the overall model.

In conclusion, our method showed competitive performance compared to the state-of-the-art MR and Hit@10 on the FB15k, WN18, FB15k-237, and WN18RR benchmark datasets. Notably, it outperforms the current best models on the benchmark dedicated to relations with the composition pattern. Contributions to Research Question RQ1 are summarized as follows:

- Provided theoretical analysis of knowledge graph embedding models and pointed out their limitation on learning different relation patterns.
- Developed MDE knowledge graph embedding method and Showed that MDE allows encoding several relational patterns.
- Suggesting limit-based loss function for knowledge graph embedding.
- Proposed an algorithm to find the limits for the limit-based loss function to use in embedding models.

- Empirically evaluated our method, which performed competitively to the state-of-the-art in the link prediction experiments.

Research Question 2 (RQ2)

Does learning network features of knowledge graphs improve the efficiency of KG embedding?

Graph feature-aware embedding of knowledge graphs has become possible after we resolved the RQ1 with a model capable of intaking more information than one feature per training sample compared to the conventional KGE methods. Based on this accomplishment, we investigated to find a solution for the research question RQ2 in Chapter 5.

We observed that the global network factors provide a new insight to better separate and distinguish the graph nodes. For instance, a representation vector for a node using feature-aware embedding includes the details like degree and page rank of that node. In addition, it indicates the node's position regarding the whole knowledge graph. Acquiring this quantifier information in knowledge graph embedding would make the embedding method more precise, allowing it better distinguish entities and better predict hidden links. Thus, we aimed to make a multi-objective formulation to perform graph feature-aware embeddings based on entities' centrality and distance features. We developed the GFA-NN embedding method based on this idea, and we investigated the advantages and limitations of the model both theoretically and empirically.

The empirical comparison of the model affirms its efficiency improvement over the state-of-the-art methods in the link prediction task. We observed a more pronounced improvement in the large-scale knowledge graph embedding experiment. The model also beats methods dedicated to inductive knowledge graph embedding, indicating it has **inductive bias**. Contributions to Research Question RQ2, in summary, are as follows:

- Proposed a score function formulation for knowledge graph graph embedding that allows encoding graph features beside relational learning.

- Provided the mathematical formulation of several graph features, normalizing and adopting these features to make them compatible with relational learning.

- Proposed the GFA-NN knowledge graph embedding method: a neural network approach for feature-aware knowledge graph embedding.

- Theoretical analyzed the learning of graph features in KGE methods, highlighting their potential and limitations.

- Empirically evaluated graph feature aware embedding of knowledge graphs, where it outperformed the state-of-the-art KGE methods in several benchmarks.

Research Question 3 (RQ3)

How can we make the evaluations of embedding models reproducible?

Evaluation methods are a primary section of scientific studies, and reproducible experiments indicate the validity of an evaluation. Therefore, the lack of reproducible evaluations for embedding models is a crucial challenge. To address the RQ3 in Chapter 6, we developed a framework based on the FAIR principles to generate read-only evaluation docker images. These executable images contain a fixed executable link prediction experiment for a knowledge graph embedding method. A link prediction investigation made by this approach is operating system-independent and device-independent. Once an experiment is initialized in such a setting, it remains unchangeable, and the later reruns repeat the exact copy of the initial experiment. In addition to this testing framework, we developed a usage guide and an extension template that we published openly on Github. Contributions to Research Question RQ3 are summarized as follows:

- Developed the Benchembedd framework that generates fixed test environments to perform reproducible link prediction evaluation of KGE models.
- Presented the framework structure and provided a base for understanding the approach.
- Provided instructions for adaptation of the work in further studies to promote the application of reproducible link prediction experiments in future works.
- Provided templates to simplify the extension and deployment of the framework to promote reproductive link prediction studies.

Research Question 4

How can we recognize a more effective embedding method on a specific relation pattern?

To achieve our research objective, we find a potential challenge of lacking an evaluation dataset and benchmark that tests embedding methods as link predictors against individual relation patterns. In particular, such an evaluation dataset is required to avoid leakage between different relation patterns. We developed an approach that extracts four relation patterns and investigates the output datasets to recognize and bypass relation pattern leakage. We applied this approach to extract **96** datasets in 4 different inductive settings from FB15k and WN18 datasets. We consequently conducted a large-scale benchmark of state-of-the-art KGE methods based on these datasets. Our experiments generate meaningful results highlighting the limitations and effectiveness of the included models on individual relation patterns and categories of inductiveness. Particularly we compared the MDE approach from Chapter 4 and the graph feature-aware GFA-NN method from Chapter 5, where we observed that the addition of the graph feature in the process of learning has effectively improved the performance of the model.

Our Contributions to target RQ4 are summarized as follows:

- Proposed a new benchmark for link prediction task that targets investigating KGE models on a single relation pattern basis.
- Proposed several datasets by classifying triplets into their respective classes according to their patterns, keeping in mind the properties of both inductive and transductive types. Therefore, we extract four categories from each class: Fully Inductive, Fully Transductive, CountBased Inductive, and Head or Tail Inductive. Each category is further divided into patterns of Symmetry, Anti-symmetry, Inverse, and Inductive, making of **32** datasets per category and **96** in total.
- Developed methods for extracting separated patterns and, in addition, made automatic emending methods to avoid data leakage between detests. The datasets are also designed based on unification to benchmark them onto different link predicting models.
- We observed a significant setback in the benchmarking of knowledge graph models. Therefore we extended the work done by [60], keeping the missing leak-free evaluations in mind.
- Generated a fair comparison to help choose the best model and dataset combination, which is especially beneficial for NLP Research.
- The previous research on benchmarking datasets was too general; we provided a tool-set to designate a specific approach according to the type of datasets.
- Explored the characteristics of the datasets that can be potential performance boosters.

To sum up our contribution, we created a reproducible evaluation environment that is user-friendly for all. We designed our benchmark datasets, keeping the ease of use in mind.

Research Question 5 (RQ5)

Does a KGE method that allows the encoding of relation patterns tackle a real-world link prediction task more effectively than a commonly applied KGE?

This research question is investigated in Chapter 8. Because knowledge graph embedding approaches can predict hidden links, they are directly employable as a recommender system. Accordingly, we designed a pipeline of link prediction based on the knowledge graph embedding model developed in Chapter 4.

The **tangible outcome** of this work was a Tweet recommending system that suggests the latest research results published by researchers to physicians according to their medical learning interests implied from their social interactions on the Twitter network. Our goal in making such a system was to show the potential of knowledge graph embedding in solving the real-life challenge of helping physicians stay up-to-date on their specialized topics.

Although the proposed approach is general, in a pre-processing step, we precisely filtered out Tweets that contain relevant research results and the sub-network of physicians and researchers in the medical science for two main reasons: first, since the model predicts based on the existing interactions, the filtered interaction network provides the base for the model to lay the predictions upon; and second, learning more diverse interactions, for instance, the interaction of physicians outside their work topic act as outlier data in the training data which decrease the performance of a KGE model.

We suggest this data filtering step as a preliminary stage for applications of KGE methods with targeted objectives because these approaches make link predictions on the knowledge graph data, which are heterogeneous. In contrast, de facto training data exclude irrelevant data for none KG-based machine learning methods. Here, data pre-filtering reduces the training time and improves the model's effectiveness.

Contributions of this Chapter can be summarized as follows:

- Proposed a pipeline for applying the MDE machine learning method as a recommender system to support the social good.
- Demonstrated a method for topic-based data extraction and processing stream data of Twitter to obtain meaningful relations and generating a social knowledge graph from the harvested data.
- Proposed a formulation of link probability for one particular link in a knowledge graph based on the average outcome of the KGE on previously learned relations.
- Experimented suggestion of connections for physicians to the latest medical study results meaningfully relevant to them on Twitter data, intending to help physicians stay up-to-date in their field.

9.2 Limitations and Future Directions

During the course of this study, we discovered three main limitations, and we report them here to be considered in future research efforts building on our findings.

The first limitation relates to modeling the hypergraphs. Instead of edges, this type of graph includes hyperedges. A hyperedge connects more than two vertices in contrast to regular edges. Although it is possible to convert hypergraphs to graphs, the link prediction problem for a hypergraph also requires considering the conversion to a regular graph; therefore, new methods try to encode the hypergraphs without converting them to graphs. We considered this direction of study out of the scope of this thesis. While our suggested embedding models do not directly model hypergraphs, our effort can be extended in this direction.

The second limitation relates to the embedding of very large-scale knowledge graphs. We experimented with embedding large-scale graphs using GFA-NN in Chapter 5. We observed that despite producing effective embeddings, the training time of the model becomes lengthy. For instance, the training time for ogbl-biokg, which comprises about 4.8 million train triples, was 12 days, running on four parallel GPUs. To improve the training time limitation, we suggest applying distributed training techniques for the embedding models, which is out of the scope of this thesis.

The third limitation is based on our study of embedding knowledge graphs with missing entities. The inductive link prediction experiments in our study in Chapter 5 cover this type of challenge. While the proposed model was not targeted for this task, it was remarkably effective due to its comprehension of graph features. A further study in this direction to find and extract relevant graph features for inductive setting is out of the scope of this thesis, and we left it out for future works.

Based on our research results and the contributions made in this thesis, we now explore prospects for future directions and opportunities that open up after our work.

- *Future of multiple objective embeddings*: In this study, we considered multiple objective learning of distance-based modeling of relations. Further study to combine newer objectives for modeling additional relation patterns is beneficial in future works. For example, the extension to encode transitivity relation patterns can be considered. For this extension, it is sufficient to involve the projection formulation of this pattern from a recent distance-based method [80] in our suggested combination method of scores. Similarly, the proposed multi-objective optimization method based on independent weights can be easily extended with supplementary scores and a new set of embedding vectors to encode multi-modal knowledge graphs.
- *Future of graph feature learning*: We investigated the embedding of centrality and positional graph features related to entities in our study. We suggest advancing this study to involve graph features describing global features of relations. We expect future studies in this direction to improve the embedding of very heterogeneous knowledge graphs that involve a significant number of relation types.
- *Future of relation pattern evaluations*: Our evaluation study on individual relation patterns produces meaningful results in spotting the weakness and competency of different models. A complimentary work can be the development of test sets with particular cases that include a calculated mix of different relation patterns. For example, the triples with a mix of relation patterns in some cases are more manageable for a KGE method to predict because the model might be able to encode only one of the patterns. Although such a model would be able to make a correct prediction in the leaked-relation pattern test, it might have trouble predicting triples with the mutually exclusive set of relation patterns. We suggest this type of evaluation for future work as we observe these combinations in real-world knowledge graphs.
- *Future of better handling the unbalanced set of negative and positive samples in KG learning*: In our study, we recognized that in most novel models, despite producing a high MRR value in the link prediction task, the MR output is high as well, which indicates these models are overfitting. The existence of numerous negative samples per positive sample is an issue in knowledge graph modeling and is the cause of the overfit in most KGE models. We targeted this problem using limit-based loss, while another study [18] aimed to solve this problem by deploying adversarial negative sampling. Both these studies generate a balanced scoring combination for negative and positive samples; however, one of the two techniques was more effective inconstantly on different datasets. We suggest combining these approaches in future works to handle the unbalanced training samples better.

9.3 Closing Remarks

Knowledge graph embedding models aim to encode graphs into representation weights and deduct their hidden relations. Learning complex knowledge graphs, however, face challenges where various relation patterns frequently appear in the network and numerous similar entities with hardly distinguishable neighborhoods making the embedding task difficult.

In this study, we proposed approaches to target these challenges and developed datasets and an evaluation framework to better recognize and compare KGE models' performance on such problems. We suggested an end-to-end machine learning workflow for embedding models and practically explored this workflow by deploying our suggested KGE method in a real-world application. More specifically, we proposed and developed:

- Proposed a multiple-objective approach for the effective embedding of knowledge graphs.
- Proposed Graph feature-aware embeddings for better encoding of knowledge graphs.
- Developed a benchmark and dataset for precise leak-free relation pattern benchmarking of link predictors.
- Developed a framework for the reproducible evaluation of embedding models.
- Investigated the application of our multiple-distance embedding method to a real-world task with benefits to the social good.

Future research can extend the datasets and proposed methodologies presented in this thesis as contributions. These efforts could provide the groundwork for highly accurate complex knowledge graph embedding. Furthermore, our thesis' contributions are already impacting the knowledge graph learning community; Several other studies are working on relation pattern learning utilizing our approaches in link prediction tasks using the methods published throughout the thesis' duration. To contribute to the research community, we openly published all mentioned thesis contributions in code, datasets, and article format.

Bibliography

[1] M. Spivak, *A comprehensive introduction to differential geometry*, vol. 1, Publish or Perish, Incorporated, 1975 (cit. on p. 1).

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “Dbpedia: A nucleus for a web of open data,” *The semantic web*, Springer, 2007 722 (cit. on p. 1).

[3] V. N. Ioannidis, X. Song, S. Manchanda, M. Li, X. Pan, D. Zheng, X. Ning, X. Zeng, and G. Karypis, *DRKG - Drug Repurposing Knowledge Graph for Covid-19*, `https://github.com/gnn4dr/DRKG/`, 2020 (cit. on p. 1).

[4] T. Gaudelet, B. Day, A. R. Jamasb, J. Soman, C. Regep, G. Liu, J. B. R. Hayter, R. Vickers, C. Roberts, J. Tang, D. Roblin, T. L. Blundell, M. M. Bronstein, and J. P. Taylor-King, *Utilizing graph machine learning within drug discovery and development*, Briefings in Bioinformatics **22** (2021), bbab159, ISSN: 1477-4054, eprint: `https://academic.oup.com/bib/article-pdf/22/6/bbab159/41087478/bbab159.pdf`, URL: `https://doi.org/10.1093/bib/bbab159` (cit. on p. 2).

[5] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, “Learning to Simulate Complex Physics with Graph Networks,” *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119, Proceedings of Machine Learning Research, PMLR, 2020 8459, URL: `http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html` (cit. on p. 2).

[6] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, “Learning hierarchical representation model for nextbasket recommendation,” *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2015 403 (cit. on p. 2).

[7] A. Sadeghi, H. A. Malik, D. Collarana, and J. Lehmann, “Relational Pattern Benchmarking on the Knowledge Graph Link Prediction Task,” *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021 (cit. on pp. 3, 25, 70).

[8] R. Damiano and A. Lieto, “Ontological representations of narratives: a case study on stories and actions,” *2013 Workshop on Computational Models of Narrative*, Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2013 76 (cit. on p. 4).

[9] T. R. Gruber and G. R. Olsen, "An Ontology for Engineering Mathematics," *Principles of Knowledge Representation and Reasoning*, ed. by J. Doyle, E. Sandewall, and P. Torasso, The Morgan Kaufmann Series in Representation and Reasoning, Morgan Kaufmann, 1994 258, URL: https://www.sciencedirect.com/science/article/pii/B9781483214528501202 (cit. on p. 4).

[10] Q. Wang, Z. Mao, B. Wang, and L. Guo, *Knowledge Graph Embedding: A Survey of Approaches and Applications*, IEEE Transactions on Knowledge and Data Engineering **29** (2017) 2724, URL: https://doi.org/10.1109/TKDE.2017.2754499 (cit. on p. 6).

[11] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, *A Survey on Knowledge Graphs: Representation, Acquisition, and Applications*, IEEE Transactions on Neural Networks and Learning Systems **33** (2022) 494, URL: https://doi.org/10.1109/TNNLS.2021.3070843 (cit. on p. 6).

[12] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding Entities and Relations for Learning and Inference in Knowledge Bases," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015 (cit. on pp. 6, 27, 33, 40, 44–46, 77).

[13] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex Embeddings for Simple Link Prediction," *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, ed. by M. Balcan and K. Q. Weinberger, vol. 48, JMLR Workshop and Conference Proceedings, JMLR.org, 2016 2071, URL: http://proceedings.mlr.press/v48/trouillon16.html (cit. on pp. 6, 27, 28, 33, 41, 44–46).

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," *Advances in Neural Information Processing Systems*, ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, vol. 26, Curran Associates, Inc., 2013 2787, URL: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf (cit. on pp. 7, 23, 26, 30, 32, 34, 41, 45, 50, 59, 65, 77, 88).

[15] A. Sadeghi, D. Graux, H. Shariat Yazdi, and J. Lehmann, "MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs," *ECAI*, 2020, URL: http://ecai2020.eu/papers/1271_paper.pdf (cit. on pp. 8, 25, 28, 32, 51, 57, 60, 77, 86, 89).

[16] L. Ehrlinger and W. Wöß, *Towards a Definition of Knowledge Graphs.*, SEMANTiCS (Posters, Demos, SuCCESS) **48** (2016) 2 (cit. on p. 17).

[17] S. Zhang, Y. Tay, L. Yao, and Q. Liu, "Quaternion Knowledge Graph Embeddings," *NeurIPS*, 2019 2731 (cit. on pp. 19, 57, 60).

[18] Z. Sun, Z. Deng, J. Nie, and J. Tang, "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019, URL: https://openreview.net/forum?id=HkgEQnRqYQ (cit. on pp. 19, 28, 77, 83, 99).

[19] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, *Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago*, Semantic Web **9** (2018) 77 (cit. on p. 19).

[20] G. Strang, G. Strang, G. Strang, and G. Strang, *Introduction to linear algebra*, vol. 3, Wellesley-Cambridge Press Wellesley, MA, 1993 (cit. on p. 22).

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*, IEEE Signal processing magazine **29** (2012) 82 (cit. on p. 23).

[22] M. Nickel, V. Tresp, and H.-P. Kriegel, "A Three-Way Model for Collective Learning on Multi-Relational Data.," *ICML*, vol. 11, 2011 809 (cit. on pp. 23, 33, 45, 88).

[23] M. D. Zeiler, *ADADELTA: An Adaptive Learning Rate Method*, CoRR **abs/1212.5701** (2012), arXiv: 1212.5701, URL: http://arxiv.org/abs/1212.5701 (cit. on pp. 23, 38, 42, 78).

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Y. Bengio and Y. LeCun, 2015, URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 23, 56).

[25] N. Gunantara, *A review of multi-objective optimization: Methods and its applications*, Cogent Engineering **5** (2018) 1502242 (cit. on p. 23).

[26] L. Zadeh, *Optimality and non-scalar-valued performance criteria*, IEEE transactions on Automatic Control **8** (1963) 59 (cit. on p. 24).

[27] A. Sadeghi, D. Collarana, D. Graux, and J. Lehmann, "Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities," *European Conference on Machine Learning and Data Mining, ECML PKDD 2021*, 2021 (cit. on pp. 25, 28, 29, 50, 82, 83).

[28] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning Entity and Relation Embeddings for Knowledge Graph Completion," *29th AAAI conference on artificial intelligence*, 2015 2181 (cit. on pp. 26, 32, 34, 50).

[29] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, ed. by C. E. Brodley and P. Stone, AAAI Press, 2014 1112, URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8531 (cit. on pp. 26, 28, 32, 34, 45, 50).

[30] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang,
"RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space," *ICLR*, 2019, URL: https://openreview.net/forum?id=HkgEQnRqYQ
(cit. on pp. 26, 32–34, 40, 41, 44–46, 50, 56, 57, 60).

[31] H. Zhu, R. Xie, Z. Liu, and M. Sun,
"Iterative Entity Alignment via Joint Knowledge Embeddings,"
Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ed. by C. Sierra, ijcai.org, 2017 4258, URL: https://doi.org/10.24963/ijcai.2017/595 (cit. on p. 26).

[32] M. Chen, Y. Tian, M. Yang, and C. Zaniolo,
"Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment,"
Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ed. by C. Sierra, ijcai.org, 2017 1511, URL: https://doi.org/10.24963/ijcai.2017/209 (cit. on p. 26).

[33] Z. Sun, W. Hu, Q. Zhang, and Y. Qu,
"Bootstrapping Entity Alignment with Knowledge Graph Embedding.," *IJCAI*, 2018 4396
(cit. on pp. 26, 37).

[34] X. Shi and Y. Xiao,
"Modeling multi-mapping relations for precise cross-lingual entity alignment," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019 813 (cit. on p. 26).

[35] M. Nickel and V. Tresp, "Tensor factorization for multi-relational learning,"
Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013 617 (cit. on p. 27).

[36] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion,"
ACM SIGKDD international conference on Knowledge discovery and data mining,
ACM, 2014 601 (cit. on pp. 27, 45).

[37] R. Socher, D. Chen, C. D. Manning, and A. Ng,
"Reasoning with neural tensor networks for knowledge base completion,"
Advances in neural information processing systems, 2013 926 (cit. on pp. 27, 45).

[38] A. Bordes, X. Glorot, J. Weston, and Y. Bengio,
A semantic matching energy function for learning with multi-relational data,
Machine Learning **94** (2014) 233 (cit. on p. 27).

[39] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl,
"Neural Message Passing for Quantum Chemistry,"
Proceedings of the 34th International Conference on Machine Learning - Volume 70,
ICML'17, JMLR.org, 2017 1263,
URL: http://dl.acm.org/citation.cfm?id=3305381.3305512 (cit. on p. 27).

[40] D. Marcheggiani and I. Titov,
"Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017 1506,
URL: http://aclweb.org/anthology/D17-1159 (cit. on p. 27).

[41] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling,
Modeling Relational Data with Graph Convolutional Networks,
arXiv preprint arXiv:1703.06103 (2017) (cit. on p. 27).

[42] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou,
"End-to-end structure-aware convolutional networks for knowledge base completion," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 01, 2019 3060
(cit. on p. 27).

[43] R. Ye, X. Li, Y. Fang, H. Zang, and M. Wang, "A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*,
International Joint Conferences on Artificial Intelligence Organization, 2019 4135,
URL: https://doi.org/10.24963/ijcai.2019/574 (cit. on p. 27).

[44] S. Vashishth, S. Sanyal, V. Nitin, and P. P. Talukdar,
"Composition-based Multi-Relational Graph Convolutional Networks," *ICLR*, 2020,
URL: https://openreview.net/forum?id=BylA_C4tPr (cit. on pp. 28, 50, 57, 60, 77).

[45] R. Kadlec, O. Bajgar, and J. Kleindienst,
"Knowledge Base Completion: Baselines Strike Back," *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*,
ed. by P. Blunsom, A. Bordes, K. Cho, S. B. Cohen, C. Dyer, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, and S. Yih, Association for Computational Linguistics, 2017 69,
URL: https://doi.org/10.18653/v1/w17-2609 (cit. on p. 28).

[46] L. Cai and W. Y. Wang, "KBGAN: Adversarial Learning for Knowledge Graph Embeddings," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*,
ed. by M. A. Walker, H. Ji, and A. Stent, Association for Computational Linguistics, 2018 1470, URL: https://doi.org/10.18653/v1/n18-1133 (cit. on p. 28).

[47] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel,
"Convolutional 2D Knowledge Graph Embeddings," *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*,
ed. by S. A. McIlraith and K. Q. Weinberger, AAAI Press, 2018 1811,
URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366
(cit. on pp. 28, 30, 41, 44–46, 60).

[48] M. M. Luqman, J.-Y. Ramel, J. Lladós, and T. Brouard, *Fuzzy multilevel graph embedding*, Pattern Recognition **46** (2013) 551, ISSN: 0031-3203, URL: http://www.sciencedirect.com/science/article/pii/S0031320312003470 (cit. on p. 28).

[49] J. You, R. Ying, and J. Leskovec, "Position-aware Graph Neural Networks," *ICML*, 2019 7134 (cit. on pp. 29, 50, 52, 53, 57–59).

[50] C. Ren, L. Zhang, L. Fang, T. Xu, Z. Wang, S. Yuan, and E. Chen, "Ontological Concept Structure Aware Knowledge Transfer for Inductive Knowledge Graph Embedding," *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, IEEE, 2021 1, URL: https://doi.org/10.1109/IJCNN52387.2021.9533852 (cit. on p. 29).

[51] K. Teru, E. Denis, and W. Hamilton, "Inductive Relation Prediction by Subgraph Reasoning," *ICML*, 2020 9448 (cit. on pp. 29, 30, 61, 72, 77).

[52] H. Zha, Z. Chen, and X. Yan, "Inductive relation prediction by BERT," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 5, 2022 5923 (cit. on p. 29).

[53] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *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)*, ed. by J. Burstein, C. Doran, and T. Solorio, Association for Computational Linguistics, 2019 4171, URL: https://doi.org/10.18653/v1/n19-1423 (cit. on p. 29).

[54] B. Oh, S. Seo, J. Hwang, D. Lee, and K.-H. Lee, *Open-world knowledge graph completion for unseen entities and relations via attentive feature aggregation*, Information sciences **586** (2022) 468 (cit. on p. 29).

[55] R. Xie, Z. Liu, H. Luan, and M. Sun, "Image-embodied Knowledge Representation Learning," *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, ed. by C. Sierra, ijcai.org, 2017 3140, URL: https://doi.org/10.24963/ijcai.2017/438 (cit. on p. 29).

[56] S. Liu, B. C. Grau, I. Horrocks, and E. V. Kostylev, "INDIGO: GNN-Based Inductive Knowledge Graph Completion Using Pair-Wise Encoding," *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, ed. by M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, 2021 2034, URL: https://proceedings.neurips.cc/paper/2021/hash/0fd600c953cde8121262e322ef09f70e-Abstract.html (cit. on p. 29).

[57] P. Wang, J. Han, C. Li, and R. Pan, "Logic Attention Based Neighborhood Aggregation for Inductive Knowledge Graph Embedding," *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, AAAI Press, 2019 7152, URL: https://doi.org/10.1609/aaai.v33i01.33017152 (cit. on p. 29).

[58] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, *Knowledge Graph Embedding for Link Prediction: A Comparative Analysis*, ACM Transactions on Knowledge Discovery from Data (TKDD) **15** (2021) 1, ISSN: 1556-472X, URL: http://dx.doi.org/10.1145/3424672 (cit. on pp. 29, 72).

[59] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, ACL, 2015 57 (cit. on pp. 30, 41, 60).

[60] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open Graph Benchmark: Datasets for Machine Learning on Graphs," *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, 2020, URL: https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html (cit. on pp. 30, 60, 61, 70, 72, 97).

[61] Y. Zhang, H. Xu, X. Zhang, X. Wu, and Z. Yang, *TRFR: A ternary relation link prediction framework on Knowledge graphs*, Ad Hoc Networks **113** (2021) 102402, URL: https://doi.org/10.1016/j.adhoc.2020.102402 (cit. on p. 30).

[62] W. Xiong, T. Hoang, and W. Y. Wang, "DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, ed. by M. Palmer, R. Hwa, and S. Riedel, Association for Computational Linguistics, 2017 564, URL: https://doi.org/10.18653/v1/d17-1060 (cit. on pp. 30, 61).

[63] S. Guan, X. Jin, Y. Wang, and X. Cheng, "Link prediction on n-ary relational data," *Proceedings of the 28th International Conference on World Wide Web (WWW'19)*, 2019 583 (cit. on p. 30).

[64] L. De Raedt, *Logical and relational learning*, Springer Science & Business Media, 2008 (cit. on p. 32).

[65] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, *A Review of Relational Machine Learning for Knowledge Graphs*, Proceedings of the IEEE **104** (2016) 11, ISSN: 1558-2256, URL: http://dx.doi.org/10.1109/JPROC.2015.2483592 (cit. on p. 32).

[66] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge base completion via search-based question answering," *Proceedings of the 23rd international conference on World wide web*, ACM, 2014 515 (cit. on p. 32).

[67] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016 192 (cit. on p. 32).

[68] T. Jiang, T. Liu, T. Ge, L. Sha, S. Li, B. Chang, and Z. Sui, "Encoding temporal information for time-aware link prediction," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016 2350 (cit. on p. 32).

[69] C. Muhao, T. Yingtao, Y. Mohan, and Z. Carlo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," *In Proceedings of IJCAI*, 2017 1511 (cit. on p. 32).

[70] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge Graph Embedding via Dynamic Mapping Matrix," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, The Association for Computer Linguistics, 2015 687, URL: https://doi.org/10.3115/v1/p15-1067 (cit. on p. 32).

[71] Q. N. Dat, S. Kairit, Q. Lizhen, and J. Mark, "STransE: a novel embedding model of entities and relationships in knowledge bases," *In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016 460 (cit. on pp. 32, 45).

[72] S. M. Kazemi and D. Poole, "SimplE Embedding for Link Prediction in Knowledge Graphs," *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, ed. by S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 2018 4289, URL: https://proceedings.neurips.cc/paper/2018/hash/b2ab001909a8a6f04b51920306046ce5-Abstract.html (cit. on pp. 32, 33, 36, 45, 48).

[73] G. Bouchard, S. Singh, and T. Trouillon, "On Approximate Reasoning Capabilities of Low-Rank Vector Spaces," *AAAI Spring Symposium Series*, 2015 (cit. on pp. 33, 44).

[74] M. Nickel, L. Rosasco, and T. A. Poggio, "Holographic Embeddings of Knowledge Graphs," *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, ed. by D. Schuurmans and M. P. Wellman, AAAI Press, 2016 1955, URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12484 (cit. on pp. 33, 41, 44).

[75] K. Hayashi and M. Shimbo,
"On the Equivalence of Holographic and Complex Embeddings for Link Prediction,"
Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers,
ed. by R. Barzilay and M. Kan, Association for Computational Linguistics, 2017 554,
URL: https://doi.org/10.18653/v1/P17-2088 (cit. on p. 33).

[76] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*,
Journal of Mathematics and Physics **6** (1927) 164 (cit. on p. 33).

[77] T. Trouillon, C. R. Dance, É. Gaussier, J. Welbl, S. Riedel, and G. Bouchard,
Knowledge graph completion via complex tensor factorization,
The Journal of Machine Learning Research **18** (2017) 4735 (cit. on pp. 33, 48).

[78] P. D. Hoff, A. E. Raftery, and M. S. Handcock,
Latent space approaches to social network analysis,
Journal of the american Statistical association **97** (2002) 1090 (cit. on p. 34).

[79] A. Bordes, J. Weston, R. Collobert, and Y. Bengio,
"Learning structured embeddings of knowledge bases," *AAAI*, 2011 (cit. on pp. 34, 42).

[80] T. Song, J. Luo, and L. Huang,
Rot-Pro: Modeling Transitivity by Projection in Knowledge Graph Embedding,
Advances in Neural Information Processing Systems **34** (2021) (cit. on pp. 34, 99).

[81] R. T. Marler and J. S. Arora, *Survey of multi-objective optimization methods for engineering*,
Structural and multidisciplinary optimization **26** (2004) 369 (cit. on p. 36).

[82] X. Zhou, Q. Zhu, P. Liu, and L. Guo,
"Learning knowledge embeddings by combining limit-based scoring loss,"
Proceedings of the 2017 ACM on Conference on Information and Knowledge Management,
ACM, 2017 1009 (cit. on p. 37).

[83] V. Vapnik, A. Y. Chervonenkis, and N. Moskva, *Pattern recognition theory*,
Statistical Learning Problems (1974) (cit. on p. 38).

[84] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu,
"Modeling Relation Paths for Representation Learning of Knowledge Bases,"
Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015,
ed. by L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton,
The Association for Computational Linguistics, 2015 705,
URL: https://doi.org/10.18653/v1/d15-1082 (cit. on p. 40).

[85] G. A. Miller, *WordNet: A Lexical Database for English*,
Communications of the ACM **38** (1995) 39, ISSN: 0001-0782,
URL: https://doi.org/10.1145/219717.219748 (cit. on pp. 41, 70).

[86] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor,
"Freebase: a collaboratively created graph database for structuring human knowledge,"
ACM SIGMOD international conference on Management of data, AcM, 2008 1247
(cit. on p. 41).

[87] D. Q. Nguyen,
An overview of embedding models of entities and relationships for knowledge base completion, CoRR **abs/1703.08098** (2017), arXiv: `1703.08098`,
URL: `http://arxiv.org/abs/1703.08098` (cit. on p. 41).

[88] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling Relational Data with Graph Convolutional Networks," *ESWC*, ed. by A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, vol. 10843, Springer, 2018 593, URL: `https://doi.org/10.1007/978-3-319-93417-4%5C_38` (cit. on p. 50).

[89] L. Cai, B. Yan, G. Mai, K. Janowicz, and R. Zhu, "TransGCN: Coupling Transformation Assumptions with Graph Convolutional Networks for Link Prediction," *K-CAP*, ACM, 2019 131 (cit. on p. 50).

[90] P. Ristoski and H. Paulheim, "RDF2Vec: RDF Graph Embeddings for Data Mining," *ISWC*, 2016 498 (cit. on pp. 50, 52).

[91] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun,
"Representation Learning of Knowledge Graphs with Entity Descriptions," *AAAI*, 2016 2659 (cit. on p. 50).

[92] N. Veira, B. Keng, K. Padmanabhan, and A. G. Veneris,
"Unsupervised Embedding Enhancements of Knowledge Graphs using Textual Associations," *IJCAI*, 2019 (cit. on p. 50).

[93] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo,
"Knowledge Graph Embedding With Iterative Guidance From Soft Rules," *AAAI*, 2018 4816 (cit. on p. 50).

[94] W. Y. Wang and W. W. Cohen,
"Learning First-Order Logic Embeddings via Matrix Factorization," *IJCAI*, 2016 2132 (cit. on p. 50).

[95] M. Lin, Q. Chen, and S. Yan, *Network in network*, CoRR **abs/1312.4400** (2013), arXiv: `1312.4400`, URL: `https://arxiv.org/abs/1312.4400` (cit. on p. 55).

[96] L. Katz, *A new status index derived from sociometric analysis*, Psychometrika **18** (1953) 39 (cit. on p. 55).

[97] T. N. Kipf and M. Welling,
"Semi-Supervised Classification with Graph Convolutional Networks," *ICLR*, 2017 (cit. on p. 57).

[98] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio,
"Graph Attention Networks," *ICLR*, 2018 (cit. on p. 57).

[99] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka,
"Representation Learning on Graphs with Jumping Knowledge Networks," *ICML*, ed. by J. G. Dy and A. Krause, 2018 5449 (cit. on p. 57).

[100] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Ç. Gülçehre, H. F. Song, A. J. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. R. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, *Relational inductive biases, deep learning, and graph networks*, CoRR **abs/1806.01261** (2018), URL: http://arxiv.org/abs/1806.01261 (cit. on p. 58).

[101] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor Factorization for Knowledge Graph Completion," *EMNLP-IJCNLP*, 2019 5188 (cit. on p. 60).

[102] T. Lacroix, N. Usunier, and G. Obozinski, "Canonical tensor decomposition for knowledge base completion," *ICML*, 2018 2863 (cit. on p. 60).

[103] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, and H. Stuckenschmidt, "Fine-Grained Evaluation of Rule- and Embedding-Based Systems for Knowledge Graph Completion," *ISWC*, 2018 3 (cit. on p. 61).

[104] F. Yang, Z. Yang, and W. W. Cohen, "Differentiable Learning of Logical Rules for Knowledge Base Reasoning," *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, ed. by I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, 2017 2319, URL: https://proceedings.neurips.cc/paper/2017/hash/0e55666a4ad822e0e34299df3591d979-Abstract.html (cit. on pp. 61, 72).

[105] A. Sadeghian, M. Armandpour, P. Ding, and D. Z. Wang, "DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs," *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, ed. by H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, 2019 15321, URL: https://proceedings.neurips.cc/paper/2019/hash/0c72cb7ee1512f800abe27823a792d03-Abstract.html (cit. on p. 61).

[106] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., *The FAIR Guiding Principles for scientific data management and stewardship*, Nature Scientific data (2016) (cit. on pp. 63, 64).

[107] A. Sadeghi, X. Shahini, M. Schmitz, and J. Lehmann, "BenchEmbedd: A FAIR Benchmarking tool for knowledge graph Embeddings," *SEMANTiCS 2021 Demo Session*, http://ceur-ws.org/, 2021 (cit. on pp. 64, 71).

[108] M. Röder, D. Kuchelev, and A. N. Ngomo, *HOBBIT: A platform for benchmarking Big Linked Data*, Data Science **3** (2020) 15, URL: https://doi.org/10.3233/ds-190021 (cit. on p. 64).

[109] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, *DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia*, Semantic Web Journal **6** (2015) 167, URL: https://doi.org/10.3233/SW-140134 (cit. on p. 70).

[110] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A Core of Semantic Knowledge," *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, Association for Computing Machinery, 2007 697, ISBN: 9781595936547, URL: https://doi.org/10.1145/1242572.1242667 (cit. on p. 70).

[111] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, ed. by J. T. Wang, ACM, 2008 1247, URL: https://doi.org/10.1145/1376616.1376746 (cit. on p. 70).

[112] Y. Wang, D. Ruffinelli, R. Gemulla, S. Broscheit, and C. Meilicke, "On Evaluating Embedding Models for Knowledge Base Completion," *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*, ed. by I. Augenstein, S. Gella, S. Ruder, K. Kann, B. Can, J. Welbl, A. Conneau, X. Ren, and M. Rei, Association for Computational Linguistics, 2019 104, URL: https://doi.org/10.18653/v1/w19-4313 (cit. on p. 70).

[113] A. Sarkar, Y. Yang, and M. Vihinen, *Variation benchmark datasets: update, criteria, quality and applications*, Database (2020), ISSN: 1758-0463, URL: https://doi.org/10.1093/database/baz117 (cit. on p. 70).

[114] T. Safavi and D. Koutra, "CoDEx: A Comprehensive Knowledge Graph Completion Benchmark," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by B. Webber, T. Cohn, Y. He, and Y. Liu, Association for Computational Linguistics, 2020 8328, URL: https://doi.org/10.18653/v1/2020.emnlp-main.669 (cit. on p. 72).

[115] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, *TUDataset: A collection of benchmark datasets for learning with graphs*, 2020, arXiv: 2007.08663 [cs.LG] (cit. on p. 72).

[116] C. Meilicke, M. Fink, Y. Wang, D. Ruffinelli, R. Gemulla, and H. Stuckenschmidt, "Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion," *International semantic web conference*, Springer, 2018 3 (cit. on p. 72).

[117] H. Wang, K. Qin, R. Y. Zakari, G. Lu, and J. Yin, *Deep Neural Network Based Relation Extraction: An Overview*, Neural Computing and Applications **34** (2022) 4781, URL: https://doi.org/10.1007/s00521-021-06667-3 (cit. on p. 72).

[118] M. Yousef, S. Jung, L. Showe, and M. Showe, *Learning from positive examples when the negative class is undetermined- microRNA gene identification*, Algorithms for molecular biology : AMB **3** (2008) 2 (cit. on p. 77).

[119] A. Sadeghi and J. Lehmann, "Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter," *Machine Learning and Knowledge Discovery in Databases*, ed. by P. Cellier and K. Driessens, Springer International Publishing, 2020 622, ISBN: 978-3-030-43823-4 (cit. on p. 86).

[120] W. Raghupathi and V. Raghupathi, *Big data analytics in healthcare: promise and potential*, Health information science and systems **2** (2014) 3 (cit. on p. 86).

[121] C. Hawn, *Take two aspirin and tweet me in the morning: how Twitter, Facebook, and other social media are reshaping health care*, Health affairs **28** (2009) 361 (cit. on p. 86).

[122] E. K. Choo, M. L. Ranney, T. M. Chan, N. S. Trueger, A. E. Walsh, K. Tegtmeyer, S. O. McNamara, R. Y. Choi, and C. L. Carroll, *Twitter as a tool for communication and knowledge exchange in academic medicine: A guide for skeptics and novices*, Medical Teacher **37** (2015) 411, URL: https://doi.org/10.3109/0142159X.2014.993371 (cit. on p. 86).

[123] N. Pemmaraju, M. A. Thompson, and M. Qazilbash, "Disease-specific hashtags and the creation of Twitter medical communities in hematology and oncology," *Seminars in hematology*, vol. 54, Elsevier, 2017 189 (cit. on p. 86).

[124] R. Brady, S. Chapman, S. Atallah, M. Chand, J. Mayol, A. Lacy, and S. Wexner, *# colorectalsurgery*, British Journal of Surgery **104** (2017) 1470 (cit. on p. 86).

[125] Y. Pershad, P. Hangge, H. Albadawi, and R. Oklu, *Social medicine: Twitter in healthcare*, Journal of clinical medicine **7** (2018) 121 (cit. on p. 86).

[126] P. Gupta, A. Goel, J. Lin, A. Sharma, D. Wang, and R. Zadeh, "WTF: The Who to Follow Service at Twitter," *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, ACM, 2013 505, ISBN: 978-1-4503-2035-1, URL: http://doi.acm.org/10.1145/2488388.2488433 (cit. on p. 86).

[127] A. Goel, A. Sharma, D. Wang, and Z. Yin, "Discovering similar users on twitter," *11th Workshop on Mining and Learning with Graphs*, 2013 (cit. on p. 86).

[128] S. Martincic-Ipsic, E. Mocibob, and M. Perc, "Link prediction on Twitter," *PloS one*, 7, Public Library of Science San Francisco, CA USA, 2017 (cit. on p. 88).

[129] W. Almansoori, S. Gao, T. N. Jarada, A. M. Elsheikh, A. N. Murshed, J. Jida, R. Alhajj, and J. Rokne, *Link prediction and classification in social networks and its application in healthcare and systems biology*, Network Modeling Analysis in Health Informatics and Bioinformatics **1** (2012) 27 (cit. on p. 88).

[130] S. F. Adafre and M. de Rijke, "Discovering missing links in Wikipedia," *Proceedings of the 3rd international workshop on Link discovery*, ACM, 2005 90 (cit. on p. 88).

[131] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, “Link prediction using supervised learning,” *SDM06: workshop on link analysis, counter-terrorism and security*, 2006 (cit. on p. 88).

[132] S. Vartak, *A survey on link prediction*, State University of New York, Binghamton, NY–13902, USA (2008) (cit. on p. 88).

[133] D. J. Watts and S. H. Strogatz, *Collective dynamics of ‘small-world’networks*, nature **393** (1998) 440 (cit. on p. 91).

[134] L. Campbell, Y. Evans, M. Pumper, and M. A. Moreno, *Social media use by physicians: a qualitative study of the new frontier of medicine*, BMC medical informatics and decision making **16** (2016) 91 (cit. on p. 92).

Appendix

APPENDIX A

List of Publications

- **Afshin Sadeghi**, Hirra Abdul Malik, Diego Collarana, Jens Lehmann. *Relational Pattern Benchmarking on the Knowledge Graph Link Prediction Task*. Conference on Neural Information Processing Systems (NeurIPS) 2021.

- **Afshin Sadeghi**, Diego Collarana, Damien Graux, Jens Lehmann. *Embedding Knowledge Graphs Attentive to Positional and Centrality Qualities*. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD) 2021, 548–564.

- **Afshin Sadeghi**, Xhulia Shahini, Martin Schmitz, Jens Lehmann *BenchEmbedd: A FAIR Benchmarking tool for Knowledge Graph Embeddings*. Demo track SEMANTiCS 2021. This paper is mainly collaborated by Xhulia Shahini in implementation and is partially collaborated by Martin Schmitz for the deployment setup.

- **Afshin Sadeghi**, Damien Graux, Hamed Shariat Yazdi, Jens Lehmann. *MDE: Multiple Distance Embeddings for Link Prediction in Knowledge Graphs*. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI) 2020, 1427–1434.

- **Afshin Sadeghi**, Jens Lehmann. *Linking Physicians to Medical Research Results via Knowledge Graph Embeddings and Twitter*. The 4th Workshop on Data Science for Social Good - SoGood 2019. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2019) 2019, 622–630.

List of Figures

1.1 Introducing the two facets of an End-to-End machine learning workflow deploying KGE-based services. 3
1.2 A knowledge graph consisting of information about the city Naples in Florida (US) and Naples in Italy is represented by nodes e_1 and e_2. The Challenges and Research Questions of the thesis are illustrated together in this example. 6
1.3 Demonstrating the connection of Research Questions to the machine learning End-to-End workflow involving link prediction on knowledge graphs. 9
1.4 Contributions to the Research Questions. 12
1.5 The position of the our Contributions in relation to steps in an End-to-End workflow for machine learning models involving link prediction on knowledge graphs. 14

3.1 A simple illustration of TransE and TransH. The Figure is from [29]. 26
3.2 An illustration of TransE and RotatE in complex plane. The Figure is from [30]. . . . 26
3.3 A visualization of the neural tensor network. Each dashed box represents an slice of the tensor, in this case there are k = 2 slices. The Figure is from [37]. 27

4.1 Geometric illustration of the translation terms considered in MDE. 37
4.2 Illustration of the possible positioning of score values for MDE_{NN} on WN18RR where the value of γ_1 and γ_2 is 2. 40
4.3 Diagrams describing the prediction of each term in MDE score for symmetric relation in a positive triple are depicted in Figure (a) and its corrupted version with the same head and tail in Figure (b). Figure (c) shows the histogram diagram of the elements of two the sum of two inverse relations, hypernym and hyponym in S_1. 42
4.4 Diagrams (a, b, c & d) show the norm of the elements in vectors r_1, r_2, r_3 and r_1+r_2-r_3 where r_3 is composed of r_1 and r_2. Here, r_1 represents /award/award_category/nominees./award/award_nominatio/nominated_for , r_2 represents /award/award_nominee/award_nominations./award /award_nomination/nominated_for and r_3 represents /award/award_winner/awards_won./award/award_honor/award_winner. 43

5.1 An example knowledge graph in which nodes e_1 and e_2 are difficult to distinguish by a KGE model only using their neighborhood information. 51
5.2 Architecture of GFA-NN. GFA-NN first pre-computes the centrality property of nodes and their distance to a set of to randomly selected reference nodes (**Left**). Then, node centrality and position embeddings attentive to position z_{v_m} are computed via scores $F_1, ..., F_k$ from the distance between a given node v_i and the reference-sets S_i which are shared across all the entities (**Top-middle**). 54

6.1 HOBBIT platform structure. BenchEmbedd extends it with evaluations and metrics for Link prediction on knowledge graphs. 65
6.2 An example of a knowledge graph with a missing link. 66
6.3 An example of demonstrated evaluation results. 67
6.4 An example of system meta-data file. 68

7.1 Example of triplet categorization based on relation pattern. 71
7.2 We define five different relational patterns to create our datasets and benchmark. . . . 73

8.1 This diagram displays a schema of an example medical professional knowledge graph. This sample of the Twitter network contains three users and their relations. It depicts Jane as a researcher that publishes Tweets about her latest papers. Paul is a physician following her updates, and Bob is a third user that follows Paul and is a physician but does not know about Jane and her publications. The orange dot line depicts a new link suggested by our proposed method. 87
8.2 A sample of the extracted Tweets about the recent medical studies. Each row shows the content of one of the extracted Tweets. 89

List of Tables

4.1 Number of entities, relations, and triples in each division. 41

4.2 Results on the Countries datasets. Results of RotatE are taken from [30] and the results of the other models are from [47]. 44

4.3 Results on WN18 and FB15k. Best results are in bold. 45

4.4 Results on WN18RR and FB15k-237. Best ones are in bold. 46

4.5 Results of each individual term in MDE on WN18RR and FB15k-237. The best results are in bold. 47

4.6 Results of MDE after 100 iterations when removing one of the terms. The best results are in bold. 47

5.1 Statistics of the data sets used in the Experiments. 59

5.2 Results on `WN18RR` and `FB15k-237`. Best results are in bold. 60

5.3 MRR Results for `ogbl-biokg`. (Results of previous models are from [60].) 61

5.4 Hit@10 results for inductive datasets. (Other models' results are from [51].) 61

5.5 Best hyperparameter setting of GPA-NN on the benchmark datasets. 62

7.1 Inductive Setting Datasets 76

7.2 Transductive Setting Datasets 77

7.3 Head-Tail Inductive Setting Datasets 77

7.4 50% Inductive Setting Datasets 77

7.5 Hit@10 and MRR results of Link Predictors on datasets extracted from FB15k. . . . 78

7.6 Hit@10 and MRR results of Link Predictors on datasets extracted from WN18. . . . 79

7.7 AUC-PR Results of Link Predictors on datasets extracted from FB15k. 80

7.8 AUC-PR Results of Link Predictors on datasets extracted from WN18. 80

7.9 Mean AUC-PR performance of the LP methods. 81

7.10 Evaluation results of GFA-NN model compared to MDE. 82

7.11 Ranking results of the LP models on the aggregate datasets. 83

8.1 Relation types in the Social Ontology 90

8.2 Class of users in the Social Ontology 90

8.3 Results on Twitter extracted dataset (TW52). Better results are in bold. 91

8.4 Mean Probability of linking to a Tweet C for users with different communities and liked Tweets 92