Representation Learning on Relational Data

Evgeniy Faerman

München 2021

Representation Learning on Relational Data

Evgeniy Faerman

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Evgeniy Faerman
aus Tashkent

München, den 27.01.2021

Erstgutachter: Prof. Dr. Matthias Schubert
Zweitgutachter: Prof. Dr. Kristian Kersting
Drittgutachter: Prof. Dr. Davide Mottin
Tag der mündlichen Prüfung: 30.04.2021

Eidesstattliche Versicherung

Hiermit erkläre ich, Evgeniy Faerman, an Eides statt, dass die vorliegende Dissertation von mir selbständig, ohne unerlaubte Hilfe gemäß Promotionsordnung vom 12.07.2011, § 8, Abs. 2 Pkt. 5, angefertigt worden ist.

Munich, 07.05.2021 Evgeniy Faerman

Acknowledgements

Throughout the last years, I've received tremendous support and encouragement from different people. First, I want to thank my supervisor and first referee for this thesis, Prof. Dr. Matthias Schubert, who gave me the freedom and trust to pursue various ideas. I am also very grateful to Prof. Dr. Kristian Kersting and Prof. Dr. Davide Mottin for their willingness to review my thesis and interesting discussion during the thesis defense.

I am especially thankful to Prof. Dr. Thomas Seidl for his support and opportunity to pursue my aspirations and implement my ideas. It was a rewarding experience to build LMU Innovation Lab and contribute to all other activities. Thank you for your trust and encouragement!

I also want to thank all my colleagues and collaborators who helped me develop this thesis's ideas. In particular, I want to thank Felix Borutta, with whom we navigated together through the most challenging first years of our PhD study. Thank you for your passion and commitment! Furthermore, I want to extend my gratitude to Max Berrendorf. I enjoyed our discussions and learned a lot from you. I am also thankful to Michael Fromm, my guide into the world of NLP, who convinced me to work on Argument Mining. My deep gratitude also goes to all students I supervised at LMU. I've learned a great deal while working with you.

Furthermore, I thank Susanne Grienberger and Franz Krojer, who always were very helpful in administrative issues.

Finally, I want to express my great appreciation to my family for their support and encouragement. In particular, I want to thank my wife, Oksana, who believes in me more than I do myself. Thank you for your patience and continuous support!

Zusammenfassung

Menschen nutzen Informationen über Beziehungen oder Interaktionen zwischen Objekten zur Orientierung in verschiedenen Situationen. So haben wir beispielsweise mehr Vertrauen in Empfehlungen die aus unserem Freundendenkreis kommen, freunden uns eher mit den Leuten an, mit denen wir bereits gemeinsame Freunde haben, oder passen unsere Meinungen nach den Interaktionen mit anderen Personen an. In vielen Anwendungen, wo die Methoden des maschinellen Lernens eingesetzt werden, haben wir Informationen darüber, wie die Objekte miteinander interagieren und oft ist diese Information von großer Bedeutung für den Anwendungsfall. Empfehlungen in den sozialen Medien, Szenenverständnis im Bereich Computer Vision oder Vorhersagen des Verkehraufkommens sind einige Beispiele, bei denen Beziehungen eine entscheidende Rolle in der Anwendung spielen. In dieser Arbeit stellen wir verschiedene Methoden vor, die in der Lage sind die Beziehungen in den Daten zu berücksichtigen und demonstrieren deren Nutzen für verschiedene Probleme.

Eine große Anzahl von Problemen, bei denen Beziehungsinformationen eine zentrale Rolle spielen, kann durch die Modellierung von Daten durch eine Graphenstruktur und durch die Formulierung der Aufgabe als Vorhersageproblem auf dem Graphen angegangen werden. Im ersten Teil der Arbeit gehen wir das Problem der Knotenklassifikation aus verschiedenen Richtungen an. Wir beginnen mit Ansätzen des unüberwachten Lernens, die sich durch die Annahmen über die Bedeutung der Beziehungen im Graphen unterscheiden. Für einige Anwendungen, wie z.B. soziale Netzwerke, ist es eine praktikable Annahme, dass dicht verbundene Knoten ähnlich sind. Wenn wir hingegen das Passagieraufkommen eines Flughafens anhand seiner Flugverbindungen vorhersagen wollen, sind ähnliche Knoten nicht unbedingt nahe beieinander im Graphen positioniert und haben eher vergleichbare Nachbarschaftsmuster. Darüber hinaus schlagen wir neuartige Methoden zur Klassifizierung und Regression in einem semi-überwachten Setting vor, in dem die Werte der Zielvariable nur für einen kleinen Teil der Knoten bekannt sind. Wir verwenden die bekannten Labels und Informationen darüber, wie die Knoten miteinander verbunden sind, um die Bedeutung der Beziehungen und ihre Auswirkung auf die endgültige Vorhersage zu lernen.

In dem zweiten Teil der Arbeit beschäftigen wir uns mit dem Problem des Graph-Matchings. Unser erster Anwendungsfall ist der Abgleich verschiedener geografischer Karten, wobei der Fokus auf der realistischen Anwendung mit verrauschten Daten liegt. Wir stellen eine robuste Methode vor, die in der Lage ist, das Rauschen in den Daten zu ignorieren. Als nächstes gehen wir das Problem des Entity Matchings in verschiedenen Wissensgraphen an.

Wir analysieren den Prozess der manuellen Datenannotation und schlagen ein sinnvolles Setting für das Problem vor. Außerdem führen wir neue Algorithmen ein, um diesen arbeitsintensiven Prozess zu beschleunigen. Darüber hinaus analysieren wir ausführlich bestehende Ansätze für Entity Matching und die empirische Auswertung, weisen auf verschiedene Mängel hin und machen mehrere Vorschläge zur Verbesserung.

Der nächste Teil der Arbeit ist der Forschungsrichtung Argument Mining gewidmet. Argument Mining beschäftigt sich mit der automatischen Extraktion und Suche von Argumenten. Wir schlagen einen neuartigen Ansatz zur Identifizierung von Argumenten vor und zeigen, wie er relationale Informationen nutzen kann. Wir wenden unsere Methode an, um Argumente in Peer-Reviews für wissenschaftliche Publikationen zu identifizieren und zeigen, dass Argumente für den Entscheidungsprozess wesentlich sind. Außerdem gehen wir das Problem der Argumentsuche an und stellen einen neuartigen Ansatz vor, der relevante und originelle Argumente für die Anfragen der Benutzer findet.

Schließlich schlagen wir einen Ansatz für Subspace-Clustering vor. Unser Verfahren kann mit großen Datensätzen umgehen und ist in der Lage neue Objekte den gefundenen Clustern zuzuordnen. Unsere Methode lernt die Beziehungen zwischen Objekten und führt das Clustering auf dem resultierenden Graphen durch.

Abstract

Humans utilize information about relationships or interactions between objects for orientation in various situations. For example, we trust our friend circle recommendations, become friends with the people we already have shared friends with, or adapt opinions as a result of interactions with other people. In many Machine Learning applications, we also know about relationships, which bear essential information for the use-case. Recommendations in social media, scene understanding in computer vision, or traffic prediction are few examples where relationships play a crucial role in the application. In this thesis, we introduce methods taking relationships into account and demonstrate their benefits for various problems.

A large number of problems, where relationship information plays a central role, can be approached by modeling data by a graph structure and by task formulation as a prediction problem on the graph. In the first part of the thesis, we tackle the problem of node classification from various directions. We start with unsupervised learning approaches, which differ by assumptions they make about the relationship's meaning in the graph. For some applications such as social networks, it is a feasible assumption that densely connected nodes are similar. On the other hand, if we want to predict passenger traffic for the airport based on its flight connections, similar nodes are not necessarily positioned close to each other in the graph and more likely have comparable neighborhood patterns. Furthermore, we introduce novel methods for classification and regression in a semi-supervised setting, where labels of interest are known for a fraction of nodes. We use the known prediction targets and information about how nodes connect to learn the relationships' meaning and their effect on the final prediction.

In the second part of the thesis, we deal with the problem of graph matching. Our first use-case is the alignment of different geographical maps, where the focus lies on the real-life setting. We introduce a robust method that can learn to ignore the noise in the data. Next, our focus moves to the field of Entity Alignment in different Knowledge Graphs. We analyze the process of manual data annotation and propose a setting and algorithms to accelerate this labor-intensive process. Furthermore, we point to the several shortcomings in the empirical evaluations, make several suggestions on how to improve it, and extensively analyze existing approaches for the task.

The next part of the thesis is dedicated to the research direction dealing with automatic extraction and search of arguments, known as Argument Mining. We propose a novel approach for identifying arguments and demonstrate how it can make use of relational

information. We apply our method to identify arguments in peer-reviews for scientific publications and show that arguments are essential for the decision process. Furthermore, we address the problem of argument search and introduce a novel approach that retrieves relevant and original arguments for the user's queries.

Finally, we propose an approach for subspace clustering, which can deal with large datasets and assign new objects to the found clusters. Our method learns the relationships between objects and performs the clustering on the resulting graph.

Contents

Publications and Declaration of Co-Authorship

Chapter 2

The chapter 2 corresponds to the following publication:

Evgeniy Faerman, Felix Borutta, Kimon Fountoulakis, and Michael W Mahoney. "Lasagne: Locality and structure aware graph node embedding". In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. **Best Student Paper Award**. IEEE. 2018, pp. 246–253. DOI: 10.1109/WI.2018.00-83

The author proposed the research idea, developed it with Felix Borutta, and discussed it with other paper co-authors. The author implemented the approach, and Felix Borutta was responsible for the evaluation. The initial draft was written by the author and Felix Borutta and revised by all authors.

Chapter 3

The Chapter 3 corresponds to the following publications:

Felix Borutta, Julian Busch*, Evgeniy Faerman*, Adina Klink, and Matthias Schubert. "Structural Graph Representations based on Multiscale Local Network Topologies". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. *equal contribution. IEEE. 2019, pp. 91–98. DOI: 10.1145/3350546.3352505

The research idea was conceptualized and developed by the author, together with Felix Borutta, Julian Busch and Matthias Schubert. Adina Kink did initial proof-of-concept and evaluation in her bachelor thesis, supervised by Felix Borutta, the author, and Julian Busch. The primary implementation was done by Felix Borutta and evaluated by Felix Borutta and Julian Busch.

Chapter 4

The Chapter 4 corresponds to the following publications:

Evgeniy Faerman, Felix Borutta, Julian Busch, and Matthias Schubert. "Ada-LLD: Adaptive Node Similarity for Node Classification Using Multi-Scale Local Label Distributions". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020). **Best Student Paper Award**, pp. 25–32

The research idea was proposed by the author and discussed with other co-authors of the paper. The author implemented the approach, and Felix Borutta was responsible for the evaluation. The author, Julian Busch and Felix Borutta wrote the initial draft and revised it together with Matthias Schubert.

Chapter 5

The Chapter 5 corresponds to the following publication:
Evgeniy Faerman, Manuell Rogalla, Niklas Strauß, Adrian Krüger, Benedict Blümel, Max Berrendorf, Michael Fromm, and Matthias Schubert. "Spatial Interpolation with Message Passing Framework". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2019, pp. 135–141. DOI: 10.1109/ICDMW.2019.00030

The research idea was proposed by the author and discussed with other co-authors of the paper. Manuel Rogalla, Niklas Strauß, Adrian Krüger, and Benedict Blümel implemented and evaluated the approach in their practical project. The initial draft was written by the author and revised by all authors.

Chapter 6

The chapter 6 corresponds to the following publication:

Evgeniy Faerman, Otto Voggenreiter, Felix Borutta, Tobias Emrich, Max Berrendorf, and Matthias Schubert. "Graph Alignment Networks with Node Matching Scores". In: *Graph Representation Learning NeurIPS 2019 Workshop*. 2019

The research idea was proposed by the author and discussed with other co-authors of the paper. The author implemented the proposed approach. Otto Voggenreiter implemented different versions of it and all baselines and evaluated all methods in his master thesis. The master thesis of Otto Voggenreiter was supervised by the author, Tobias Emrich and Felix Borutta. The manuscript was written by author and Felix Borutta and improved by all co-authors of the paper.

Chapter 7

The chapter 7 corresponds to the following publications:

Max Berrendorf*, Evgeniy Faerman*, and Volker Tresp. "Active Learning for Entity Alignment". In: *Advances in Information Retrieval*. *equal contribution. Springer International Publishing, 2021, pp. 48–62. DOI: 10.1007/978-3-030-72113-8_4

The research idea was developed and conceptualized by author and Max Berrendorf and discussed with Volker Tresp. Max Berrendorf did the main part of the implementation, and the author implemented a few proposed heuristics. The author and Max Berrendorf wrote the manuscript.

Chapter 8

The chapter 8 corresponds to the following publications:

Max Berrendorf, Evgeniy Faerman, Laurent Vermue, and Volker Tresp. "Interpretable and Fair Comparison of Link Prediction or Entity Alignment Methods with Adjusted Mean Rank". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020), pp. 363–366

The research idea was initially proposed by Max Berrendorf, developed and conceptualized by author and Max Berrendorf, and discussed with other co-authors. Max Berrendorf did the implementation. The author and Max Berrendorf wrote the manuscript.

Chapter 9

The chapter 9 corresponds to the following publication:

Max Berrendorf, Evgeniy Faerman, Valentyn Melnychuk, Volker Tresp, and Thomas Seidl. "Knowledge graph entity alignment with graph convolutional networks: Lessons learned". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 3–11. DOI: 10.1007/978-3-030-45442-5_1

The research idea was developed and conceptualized by author and Max Berrendorf and discussed with other co-authors. The author and Max Berrendorf analyzed the original implementation, and Max Berrendorf did the reimplementation. Valentin Melnychuk ran the experiments. The manuscript was written by the author and Max Berrendorf and revisited by all co-authors.

Chapter 10

The chapter 10 corresponds to the following publication:

Max Berrendorf, Ludwig Wacker, and Evgeniy Faerman. "A Critical Assessment of State-of-the-Art in Entity Alignment". In: *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 18–32. DOI: 10.1007/978-3-030-72240-1_2

The research idea was developed and conceptualized by author and Max Berrendorf and discussed with Ludwig Wacker. Max Berrendorf and Ludwig Wacker did the implementation and evaluation. The author and Max Berrendorf wrote the manuscript.

Chapter 11

The chapter 11 corresponds to the following publication:

Michael Fromm, Evgeniy Faerman, and Thomas Seidl. "TACAM: Topic And Context Aware Argument Mining". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2019, pp. 99–106. DOI: 10.1145/3350546.3352506

The author proposed the research idea, developed and conceptualized it with Michael Fromm, and discussed it with Thomas Seidl. Michael Fromm did the implementation and evaluation. The author and Michael Fromm wrote the manuscript.

Chapter 12

The chapter 12 corresponds to the following publication:

Fromm Michael, Faerman Evgeniy, Berrendorf Max, Bhargava Siddharth, Qi Ruoxia, Zhang Yao, Dennert Lukas, Selle Sophia, Mao Yang, and Seidl Thomas. "Argument Mining Driven Analysis of Peer-Reviews". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021

The research idea was initially proposed by Michael Fromm, developed and conceptualized by the author, Michael Fromm, and Max Berrendorf and discussed with co-authors. Siddharth Bhargava, Ruoxia Qi, Yao Zhang, Lukas Dennert, Sophia Selle, and Yang Mao implemented and evaluated the approach in the practical project. The author, Michael Fromm and Max Berrendorf wrote the manuscript.

Chapter 13

The chapter 13 corresponds to the following publication:

Michael Fromm, Max Berrendorf, Sandra Obermeier, Thomas Seidl, and Evgeniy Faerman. "Diversity Aware Relevance Learning for Argument Search". In: *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 264–271. DOI: 10.1007/978-3-030-72240-1_24

The research idea was proposed by the author, developed and conceptualized by the author and other co-authors. Michael Fromm, Max Berrendorf, and Sandra Obermeier did the implementation. The author, Michael Fromm, Max Berrendorf, and Sandra Obermeier wrote the manuscript.

Chapter 14

The chapter 14 corresponds to the following publication:

Julian Busch, Evgeniy Faerman, Matthias Schubert, and Thomas Seidl. "Learning Self-Expression Metrics for Scalable and Inductive Subspace Clustering". In: *NeurIPS 2020 Workshop on Self-Supervised Learning: Theory and Practice* (2020). arXiv: 2009.12875 [cs.LG]

The initial idea was proposed by Julian Busch and conceptualized and developed by the author and Julian Busch and discussed with other co-authors. Julian Busch did the implementation and evaluation, and he also wrote the manuscript.

Chapter 1

Introduction

The field of Machine Learning (ML) deals with the development of algorithms that should be able to *learn* from previous experience, where the experience is expressed in the form of some training dataset. Given data and a task, the goal is to obtain (train) a model capable of making decisions about unseen cases generated by the same process as the training data but are not part of it. The distinguishing characteristic of ML is that in the process of training, a model has to recognize and select patterns in the data leading to the correct decisions by itself, instead of being explicitly programmed with a set of rules. While learning a model is the algorithm's responsibility, an ML practitioner's task is to design the right model and training procedure for the problem at hand. As stated in the well known *No Free Lunch* theorem Wolpert 1996; Wolpert and Macready 1997[1] no single algorithm or model works well for all possible problems in all possible scenarios. ML models differ based on expectations they have about the data, and therefore some hypothesis about the task has to be defined in the process of model selection. The set of assumptions affecting the prioritization of some solution approaches over others independent of concrete samples in the training data is known as *inductive bias* Mitchell 1980. Dependent on the task, various assumptions can be made based on the knowledge about the problem's domain, data generation process, or structural dependencies in the data. The usefulness of the selected model depends on the correlation of the made assumptions with reality.

In this thesis, our focus lies on the problems involving data with explicit *relational* structure, and we introduce and analyze methods having *relational inductive bias* Battaglia et al. 2018. A relation denotes any interaction between entities we need to make decisions about, or between their components. For example, in a social network, a relation may encode a friendship link, in a molecular graph, a bond between different atoms, and in a street graph, a relation between intersections can encode the information that a street directly connects them. Throughout the thesis, the essential assumption we make is that the relations' information is relevant to the problems we want to solve. Therefore, we make design choices enabling models to take the relational structure of the data into account. The backbone for most approaches discussed in the thesis is built by the Artificial Neural

[1] also known as a Law of Conservation for Generalization Performance in ML Schaffer 1994

Networks, which have been proven as an efficient tool for the *Representation Learning* Bengio, Courville, and Vincent 2013. The idea behind representation learning is that the feature vector describing a entity of interest in the raw form is presented to the model. The model maps it to the compact vector representation called distributed representation[2] or embedding. In the learning process, a model learns to extract concepts relevant for the target function from the input data and to encode it by a vector representation. The inductive bias in representation learning approaches is mostly encoded by the selection and composition of neural network modules or by the target task for the unsupervised methods. In contrast, *feature engineering* based approaches are designed manually to extract relevant information from the raw input based on the domain knowledge and use it to build the model's input features.

In this chapter, we aim to introduce research directions covered in the thesis, provide relevant context for understanding current state-of-the-art methods, and outline our contributions. In section 1.1 we describe our approaches' central components to give the reader a better understanding of the methods introduced in this thesis. Therefore, we start with the description of the Approximated Personalized PageRank, which we apply in several approaches to describe the node's neighborhood in the graph. Next, we describe the analog of spectral analysis on graphs and explain the effect of applying Personalized PageRank as a filter from a spectral perspective. Afterward, we provide a detailed overview of the family of methods known as Message Passing Networks, which are also part of several approaches later discussed in the thesis. After that, we overview problems addressed in the thesis and discuss earlier methods proposed to solve them in the next sections. In section 1.2, we motivate the problem of node classification in homogeneous graphs and describe unsupervised and semi-supervised approaches proposed to solve it. In section 1.3, we describe the problem of Graph Matching and focus on representation learning approaches for the inductive setting and matching of entities in different Knowledge Graphs. Moreover, in section 1.4, we introduce the field of Argument Mining and survey the current state of this research direction. Finally, in section 1.5, we describe the research questions addressed in the thesis in more detail and discuss the advantages of proposed approaches in their context.

[2]There is a many-to-many relationship between concepts in the data and dimensions in distributed representation. Each dimension is employed for the description of different concepts, and a combination of different dimensions describes each concept

1.1 Preliminaries

Whenever we refer to graph we denote it by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the set of vertices or nodes with $|\mathcal{V}| = \mathrm{N}$ and $\mathcal{E}$ set of edges with $|\mathcal{E}| = \mathrm{K}$. We use $\mathcal{A}$ to denote a binary or weighted adjacency matrix and $\mathcal{D}$ is the diagonal matrix with $\mathcal{D}_{ii} = \sum_{j=0}^{N} \mathcal{A}_{ij}$. If nodes or edges have attributes we denote them by $\mathcal{X} \in \mathbb{R}^{\mathrm{N} \times d}$ in case of node attributes and edge attributes are denoted by $E \in \mathbb{R}^{\mathrm{K} \times s}$. We use lowercase letters to denote vectors, e.g. x_i denotes attributes of node i and $e_{i,j}$ attributes of edge between i and j. $\mathbb{N}_\mathrm{j}$ is used to denote the neighborhood of the node j, and the definition of a neighborhood depends on the approach.

1.1.1 Approximated Personalized PageRank

The *PageRank* algorithm Brin and Page 1998; Page et al. 1999 was developed by the founders of Google to prioritize the results retrieved by a web search engine for a user query. The PageRank method's core assumption is that a web page's relevance correlates with its degree of popularity, and more popular websites should be returned first. Therefore the algorithm analyzes how web pages link to each other and assigns an importance score to each webpage, describing a probability to be visited by a random surfer, who randomly clicks on hyperlinks. The algorithm is not restricted to the web search application, and since the introduction, it was successfully applied to various applications requiring node importance computation in directed or undirected graphs Gleich 2015. In general, the vector with the PageRank scores is the solution of the linear system:

$$pr(\alpha, s) = \alpha s + (1 - \alpha) pr(\alpha, s) W$$

The transition matrix W, computed as $W = 0.5(I + \mathcal{D}^{-1}\mathcal{A})$, is the *lazy* variant of the random walk transition matrix, where in each step, the walk is continued or stopped at the current node with the same probability. The constant α denotes the teleportation probability; in each step, the random walk can be restarted from one of the starting nodes with the probability α or continue the walk with the probability $(1 - \alpha)$. The vector s in the equation contains the probability for each node to be the first node in random walk and, therefore, a teleportation target. The vector s is assigned with the same probability $\frac{1}{N}$ for each node, when a global importance score should be computed.

The algorithm known as *Personalized PageRank* (PPR) computes the relevance of graph nodes from the point of view of a few starting vertices. The computation of PPR is similar to PageRank. The main difference is that the vector s is sparse, and the whole teleportation probability distribution is concentrated only on starting nodes. The resulting PPR vector score describes the probability of being visited by a random walk when it is started from the starting nodes. PPR was successfully applied for various tasks on a graph, such as detecting graph communities Andersen, Chung, and Lang 2006; Kloumann and Kleinberg 2014 or linking entities recognized in the text to the knowledge base Pershina, He, and Grishman 2015. In this thesis, we use PPR to obtain the single vertex's relevant neighborhood and interpret the PPR probability as the neighbors' relevance score.

There is a non-zero probability to visit nodes situated in the same connected component by a random walk, and therefore the PPR is often characterized by long-tail distribution. However, in most real-life graphs, only a small portion of the graph contains relevant neighborhood for each node Leskovec et al. 2009, and PPR vectors can be well approximated by sparse vectors Nassar, Kloster, and Gleich 2015. In this thesis, we use the *push* algorithm from Andersen, Chung, and Lang 2006 to compute the sparse *Approximated Personalized PageRank* (APPR), where the hyperparameter controls the sparsity.

1.1.2 Spectral interpretation

The field of *Digital Signal Processing* (DSP) has had a significant impact on the development of graph-based methods in the last years. A large body of research in DSP is dedicated to signal representation. The goal is to represent a signal on some new basis from a pre-specified dictionary Rubinstein, Bruckstein, and Elad 2010, where it can be analyzed or processed more effectively and efficiently. The signals studied by DSP are usually temporally or/and spatially distributed or, more generally speaking, are sampled on regular grids (signals are often represented as functions of time or location). The primary approach is to transform the signal into the frequency domain by the *Fourier Transform* (FT). The FT represents signal by the weighted sum of cosine and sine waves of different frequencies. E.g. for the 1-dimensional time signal $x(t)$ there is a single weight $X(w)$ in the frequency domain, which indicates the presence of the corresponding frequency w in the original signal [3]:

$$x(t) = \int_{-\infty}^{\infty} X(w)e^{-iwt}dt \approx \sum_{-\infty}^{\infty} X(w)e^{-iwt}$$

While small values of w correspond to the smooth signals, which change slowly over time, signal with higher frequency changes more rapidly. FT can be used to analyze the signal spectrum or perform various operations with the signal since there are cheaper operations in the frequency domain. Afterward, the signal can also be transformed back to the original domain. The basis sine and cosine functions form an orthogonal basis in the function space.[4]

The *Laplace Operator* or *Laplacian* ∇^2 of a function in Euclidean space denotes the average difference between the function value at some point and points on the infinitesimal sphere around it. The FT basis functions are also eigenfunctions [5] of Laplacian and the corresponding frequencies are the eigenvalues. Based on this fact, the analog of FT is defined on Riemannian Manifolds Canzani 2013 and graphs Shuman et al. 2013; Stankovic,

[3] By Euler's formula a complex exponential can be represented as by sum of cosine and sine functions $e^{iwt} = cos(wt) + i * sin(wt)$.

[4] Two functions are called orthogonal if the integral of their product is zero $\int_{-\infty}^{\infty} e^{-itw_n} e^{itw_m} dt = 0, w_n \neq w_m$.

[5] An eigenfunction of an operator is a function such that the application of an operator on eigenfunction results in eigenfunction times constant Sherrill 2001

Dakovic, and Sejdic 2017 to represent signals on manifolds or graph vertices by the combination of eigenvectors of corresponding Laplacian. The Laplacian matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$[6] can be interpreted as a discrete version of the Laplacian operator. Applied to some signal $\mathcal{X} \in \mathbb{R}^{N\times 1}$ for the node i it results in $(\mathcal{L}\,X)_i = \sum_{j\in\mathbb{N}_i} a_{ij}(X_i - X_j)$. The eigendecomposition of Laplacian can be defined as $\mathcal{L} = U\Lambda U^T$, where U is the matrix with the eigenvectors and Λ is the diagonal matrix with the eigenvalues. Therefore, the FT on a graph is defined as $\hat{X} = U^T X$ and the signal reconstruction back to vertex domain as $X = U\hat{X}$. Correspondingly, the filtering operation [7] $g(\Lambda)$ is defined as $Ug(\Lambda)U^T X$. A polynomial filter $g(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k$ is commonly used in the literature Hammond, Vandergheynst, and Gribonval 2011; Defferrard, Bresson, and Vandergheynst 2016 because it is localized in the vertex space[8]. The Laplacian's eigenvalues can be interpreted as frequencies of the corresponding eigenvectors, smooth signals [9] are mainly mapped to eigenvectors with smallest eigenvalues Shuman et al. 2013. The Personalized Page Rank (PPR) transition matrix can be computed from the graph Laplacian using polynomial filters in frequency space. The corresponding filter amplifies signals mapped to the vectors with small eigenvalues and suppresses high eigenvalues Klicpera, Weißenberger, and Günnemann 2019.

1.1.3 Message Passing Neural Networks

Graph Neural Networks (GNNs) is a family of models for the representation learning on graphs. They can be designed to learn representations of nodes, edges, or whole graphs. A GNN expects an entire graph with associated node features and optional edge features as input and outputs representations of entities required for the use-case. For the most part, GNNs are applied to graphs with node features. On the graphs without node attributes, GNNs can be used with learnable node embeddings[10].

A GNN model consists of several GNN layers and output layer(s) dependent on the application. A typical GNN layer performs feature transformation by a function shared between nodes and consecutively aggregates transformed features of the node and node's neighborhood into a single vector representation. E.g. a single layer of the popular GCN Kipf and Welling 2016a model is defined as:

$$H^{t+1} = ReLU(\widehat{\mathcal{A}} H^t W) \tag{1.1}$$

where $\widehat{\mathcal{A}}$ denotes the normalized adjacency matrix with the self-loop[11], H^t is the matrix with node representations after layer t, with $H^0 = \mathcal{X}$, and W is the matrix with learned

[6] there is also normalized version is defined as $\mathcal{D}^{-\frac{1}{2}}\,\mathcal{A}\,\mathcal{D}^{-\frac{1}{2}}$, both versions are used interchangeably

[7] filtering in frequency domain corresponds to the discounting or amplifying parts of signal associated with different frequencies.

[8] since $\mathcal{L}^k = U\Lambda^k U^T$

[9] In the vertex domain, the signal is considered smooth if it has similar values at neighboring nodes

[10] There are some restrictions when using GNN with parametrized embeddings, e.g., it is necessary to make sure, that all embeddings become gradient updates, cf. 1.2.2

[11] $\widehat{\mathcal{A}} = \tilde{\mathcal{D}}^{-\frac{1}{2}}\tilde{\mathcal{A}}\tilde{\mathcal{D}}^{-\frac{1}{2}}$, with $\tilde{\mathcal{A}} = \mathcal{A} + I$ and $\tilde{\mathcal{D}}_{ii} = \sum_j \tilde{\mathcal{A}}_{ij}$

transformation parameters. This architecture resembles *convolutional layer* LeCun et al. 1999 with the neighborhood defined by a graph structure instead of proximity on the pixel grid. Some of the early GNN approaches Bruna et al. 2013; Mikael Henaff 2015; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016a were originally proposed as a generalization of convolutional networks on graphs, where the main idea is to define the convolution in the spectral domain. [12] *Geometric Deep Learning* Bronstein et al. 2017 studies the generalization of convolution to the non-euclidean domains such as graphs or manifolds.

Another perspective can be seen from the expressing GNN approaches in *Message Passing Neural Networks* (MPNN) Gilmer et al. 2017 framework. MPNN is an intuitive abstraction for the GNN which defines a GNN layer by a few basic operations. It allows easy comparison of different approaches by comparing the concrete realizations of these operations. It turned out that MPNN schema provides a practical programming model for a GNN library Fey and Lenssen 2019. In essence, most of the current GNN approaches perform following operations in each layer: [13]

$$m_{i->j}^{t+1} = M(h_i^t, h_j^t, e_{i,j}), \forall i \in \mathbb{N}_j \tag{1.2}$$

$$m_j^{t+1} = Aggr(m_{i->j}^{t+1}) \tag{1.3}$$

$$h_j^{t+1} = U_t(m_j^{t+1}, h_j^t) \tag{1.4}$$

We call vertices with outgoing links *senders* and nodes with incoming links are denoted as *targets*. In the case of undirected graph each edge is represented by two edges in both directions. The message function M in the equation 1.2 is utilized by each sender node to create a message. The weights of the function M are shared between nodes, and it expects sender's hidden representation h_i^t as input. Optionally, the representation of the target h_j^t or edge features can also be considered. In the next step, it can be imagined that messages are sent to the neighbors, and the function $Aggr$ combines all messages for each node to the single representation. Finally, the shared update function U (1.4) is responsible for the updating node's previous vector and outputs the final node representation. For example, in the GCN model described in equation 1.1, the message function M performs linear transformation with the weight matrix W. All incoming messages are summed up in the aggregation step and each message is weighted according to the corresponding entry in the matrix $\widehat{\mathcal{A}}$ [14]. The update function U in the GCN model is an identity function, and it outputs aggregated message vector as a new node representation.

Comparison of GNN approaches

Meanwhile, a vast number of different GNN approaches is proposed in the literature recently. In the following, we discuss essential ideas by comparing the realizations of MPNN

[12] According to the Convolution Theorem convolution of two functions can be performed in the Fourier domain by elementwise product of their transforms. 1.1.2 describes Fourier transformation on graphs

[13] Note, that we extend original definition to accompany more recent approaches

[14] in the GCN model the weight for message sent from i to j is $\frac{\mathcal{A}_{ij}}{\sqrt{D_{ii}D_{jj}}}$

functions. Note that we include our model presented in chapter 4 to this comparison since it also can be formulated in the MPNN framework.

Message Function Regarding the message creation and dispatching, there are two main differences among various approaches. First, it is the type of transformation performed by the function M. Furthermore, approaches differ in the definition of the neighborhood of the node.

- Feature Transformation: The majority of approaches create a message by a linear transformation of node representation from the previous layer, and non-linearity function is applied after the aggregation step, e.g., Kipf and Welling 2016a; Kipf and Welling 2016b; Veličković et al. 2017. However, simplified versions demonstrate that good results can also be achieved without feature transformation Wu et al. 2019a; Klicpera, Bojchevski, and Günnemann 2018; Thekumparampil et al. 2018. The possible explanation is that feature propagation reduces the noise alternatively to a low-pass filter in the frequency domain NT and Maehara 2019. Furthermore, there is a method utilizing the combination of transformed and original representations Chen et al. 2020.

- Definition of the neighborhood: In each layer, source nodes send messages to the neighbors. The majority of the GNN models send messages to the direct neighbors in each iteration, e.g., Kipf and Welling 2016a; Veličković et al. 2017; Thekumparampil et al. 2018. The information from remote neighbors flows in consecutive iterations through direct neighbors. However, there are also approaches sending messages directly to the indirect neighbors Klicpera, Weißenberger, and Günnemann 2019; Defferrard, Bresson, and Vandergheynst 2016; Atwood and Towsley 2016; Thekumparampil et al. 2018; chapter 4.

Aggregation An aggregation function expects a set of unordered incoming messages as input and outputs a single vector representation. An important requirement for an aggregation function is the *invariance* to permutations[15]. Following aggregation functions are applied by different MPNN approaches:

- (Weighted) MEAN: Weighted average is a popular choice for the aggregation function. Weighting controls the importance of the message for the target node and therefore many models make use of it to model various inductive biases. In general, it can be distinguished between following weighting schemes:

  - Fixed Weights: Weights stay fixed in the course of model training. E.g. the GCN model uses weights provided with the data or uses equal weights for unweighted graphs and decreases the effect of messages from high degree nodes Kipf and Welling 2016a. Alternatively, methods sending messages to k-hop neighbors

[15] $f(x_1, x_2, \cdots, x_n) = f(x_{\pi(1)}, x_{\pi(2)}, \cdots, x_{\pi(n)})$ for any permutation π

directly often use random walk probability for weighting. Therefore, the received message's weight is proportional to the probability of visiting the source by a random walk Atwood and Towsley 2016; Duvenaud et al. 2015.

  - Adjustable: Personalized Page Rank (PPR) based methods Klicpera, Bojchevski, and Günnemann 2018; Klicpera, Weißenberger, and Günnemann 2019; chapter 4 also weight messages according to the random walk probability and weights do not change in the training process. The special feature of PPR based approaches is that locality and sparsity of each node's neighborhood are controlled by teleportation probability and approximation hyperparameters. Therefore, the best hyperparameter setting can be selected using the validation set. In chapter 4, we present an approach that exploits PPR with different teleportation parameters in the message passing process and learns how to combine them optimally.
  - Attention: Attention-based approaches aim to learn the optimal weight for each message in each layer. Therefore they utilize an additional function that expects current representations of sender and target nodes and optionally an embedding of the edge. The attention function outputs weight for a message, and message scores for each target node are usually normalized with the softmax function. Initially, the attention mechanism was proposed for the sequence processing and it drastically improved results for the machine translation task Bahdanau, Cho, and Bengio 2014. Nowadays, attention based transformer architecture Vaswani et al. 2017 became state-of-the art for different types of structured data Brown et al. 2020; Parmar et al. 2018; Boes and Van hamme 2019. Conceptually, attention-based graph models Veličković et al. 2017; Thekumparampil et al. 2018 are very similar to transformer architecture. The relational bias is realized by the restriction that nodes in the graph models are allowed to attend only to the neighbors in each layer.
  - Graph Generation: Graph Generation approaches Elinas, Bonilla, and Tiao 2019; Qasim et al. 2019; Wang et al. 2019; Kazi et al. 2020 learn to generate a graph with corresponding edge weights and to perform message passing on it in the same end-to-end training procedure. The goal is to learn unknown graph structure, although some pre-existing graphs can be used as a prior. The main difference to attention-based approaches is that they determine weights for pre-specified edges, whereas Graph Generation approaches first have to decide which edges exist in the graph. Furthermore, in the case of attention, weights are recomputed in every layer.

- MAX: MAX function is a popular choice for spatial pooling in convolution networks and it is also used in GNNs Hamilton, Ying, and Leskovec 2017; Veličković et al. 2019.

- SUM: The usage of MEAN and MAX pooling functions leads to the loss of structural information such as node degree. Therefore, the SUM aggregator was proposed as

more powerful aggregation function Xu et al. 2018a; Morris et al. 2019. However, more expressive models do not necessarily lead to better performance on downstream tasks Dwivedi et al. 2020 and have worse generalization property Veličković et al. 2019. To remedy this shortcomings it was proposed to use MEAN scaled by degree-scaler function instead of SUM Corso et al. 2020. Degree-scaler is some injective function that expects a node degree as input.

- LSTM: Although *LSTM* method is not permutation invariant it was proposed to use it on random permutation of messages as an more expressive aggregation function Hamilton, Ying, and Leskovec 2017

- Set Embedding: The methods proposed for the learning of invariant set representations are directly applicable to aggregation problem in GNNs. *DeepSet* Zaheer et al. 2017 proves that the function of the form $f(X) = MLP_\theta(\sum_{x_i \in X} MLP_\delta(x_i))$ for the set X is universal set approximator. Another approach for learning of set representations called *Janossy Pooling* uses more expressive permutation-sensitive functions and approximates the average of function outputs, when it is applied on all possible reorderings Murphy et al. 2018.

- Combination: There are approaches using combinations of aggregation functions. According to Dehmamy, Barabási, and Yu 2019 combination of MEAN and SUM aggregators leads to superior performance for distinguishing between different graph models. In Corso et al. 2020 authors further combine different aggregation functions with degree scalers. Another combination approach is presented in Li et al. 2020, where softmax with temperature is applied dimensions-wise to obtain weight for each dimension of each neighbor. It can be seen as an interpolation between MAX and MEAN aggregators, where dependent on the temperature, the resulting aggregation function is closer to one of them.

Update The GNN layer's output is produced by an update function that computes node representation from aggregated messages and node representation from the previous layer. An important distinguishing feature of different approaches is how the representation of a target node is integrated into the final representation:

- Self Link: Node sends a message to itself, and it is the same message as sent to the neighbors. In this case *identity* is often used as update function and the vector with aggregated messages becomes new node representation Kipf and Welling 2016a; Veličković et al. 2017.

- Extra treatment: The common procedure to make the node's information more present in the final layer output is to use extra transformation for the own representation. Transformed node representation can be added to the representation of aggregated messages Li et al. 2020; Xu et al. 2018a; Morris et al. 2019; Cangea et al. 2018; Duvenaud et al. 2015. Note, that such an update resembles *skip-connection*, a

technique for the training of deep CNN networks He et al. 2016[16]. Another possible update is concatenation Hamilton, Ying, and Leskovec 2017; Wang et al. 2019 with the optional transformation of the resulting vector, e.g., to reduce the dimensionality. An alternative update approach utilizes a sequence model (e.g., LSTM) as an update function and to share function parameters between nodes Li et al. 2016. The message vector in each layer is used as the input to the sequence function, and hidden representation is taken as a node output.

[16]There are also other types of skip-connections in GNNs. E.g. skip connection to the representation of the first layer Chen et al. 2020, or all previous layers Fey 2019. An alternative to skip-connection is to aggregate representations from all layers for the final representation Xu et al. 2018b

1.2 Node Classification

Various practical problems can be addressed by representing the data as a graph and formulating a task as a node classification. For example, a social network can be modeled as a graph, where nodes represent people, and edges indicate friendships. In such a social graph, class labels may come in many forms: demographic data, personal interests, or even detection of misbehavior Bhagat, Cormode, and Muthukrishnan 2011. Another example is predicting protein function based on its interaction with other proteins in biological networks Zitnik and Leskovec 2017. This section discusses methods addressing node classification on homogeneous networks, where nodes may have only class label attributes. Therefore, the algorithm has access only to the interaction information and labels for a subset of labeled nodes.

In the following, we present two lines of work for the node classification. First, we discuss the unsupervised learning of node embeddings with different inductive biases. It is a two-step approach. In the first step, an embedding is trained for each node in the graph using some auxiliary tasks. Afterward, these embeddings are used as input to the classification model. In the second part of the section, we discuss models learning label assignment directly, either by propagating label information across the graph or end-to-end learning.

1.2.1 Unsupervised Learning Graphs

Unsupervised Representation Learning aims to learn useful representations of entities of interest without signal provided by humans for the final task of interest. It is also often called self-supervised learning since the learning procedure is often similar to supervised learning and the major difference is that the learning signal is derived automatically. Thus, the main objective is often to find a suitable task which provides useful supervision and enough training data. It often requires expertise in the corresponding domain. For example, state-of-the-art approaches for unsupervised learning of image representations aim to learn consistent representations of the same image across different patches or augmentations Hjelm et al. 2018; Chen et al. 2020. It turned out that in the domain of Natural Language Processing (NLP), the learning with simple Language Modeling task results in powerful models which generalize well to other tasks with little or even without task-specific supervision Devlin et al. 2018; Radford et al. 2019. In the graph domain, there are two important paradigms for the unsupervised learning of the node representations.

Homophily The first family of approaches is based on the *homophily* assumption. Homophily is the phenomena observed in many real-life networks McPherson, Smith-Lovin, and Cook 2001; Himelboim et al. 2016 that nodes modeling similar entities tend to connect to each other. Therefore, homophily based approaches (section 2) Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016; Tang et al. 2015; Cao, Lu, and Xu 2015; Wang, Cui, and Zhu 2016; Cao, Lu, and Xu 2016; Abu-El-Haija et al. 2017; Tsitsulin et al. 2018 aim to exploit this property and try to create similar (smooth) representations in the

embedding space for the nodes closely connected in the graph. The general procedure in most approaches is the same. In the first step, the method determines each node's relevant neighborhood, and the retrieved neighborhood defines the *context* of the node. The context determines the position of the node representation relative to other nodes in the embedding space. Therefore, the key distinctive features of different approaches are the specification of neighborhood and assignment of relevance to the single neighbors. There are approaches which take only neighbors from the first k hops into account and consider neighbors from the same hop neighborhood as equally important, e.g., Wang, Cui, and Zhu 2016. The hop neighborhood denotes the neighbors with the same shortest-path distance, e.g., 1-hop are direct neighbors, and 2-hop are neighbors of the neighbors, not in the 1-hop of the target node itself. Another family of methods consists of the random-walk approaches Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016; Abu-El-Haija et al. 2017; Cao, Lu, and Xu 2015; Cao, Lu, and Xu 2016 which rely on different types of random-walks or directly compute random walk probabilities to obtain the context of the target node. Therefore, the neighbor's relevance is determined by the probability of visiting it when a random-walk is started from the target node. Although these methods can do more fine-granular prioritization of neighbors, as we demonstrate in chapter 2, they have the common problem that non-relevant neighbors are heavily taken into consideration. We show how this problem can be addressed by using Approximated Page Rank.

Each node's context can be depicted in the node context co-occurrence matrix, where each value denotes the importance of the context for the target node. The graph adjacency matrix can also be seen as a co-occurrence matrix Ahmed et al. 2013, where the edge weight denotes the importance. More advanced methods use powers of the Laplacian Cao, Lu, and Xu 2015; Qiu et al. 2018 or counts of random-walk visits Abu-El-Haija et al. 2017. Some methods explicitly factorize the co-occurrence matrix into the product of representations of target nodes and context embeddings to learn the embeddings Cao, Lu, and Xu 2015; Abu-El-Haija et al. 2017; Qiu et al. 2018. An alternative approach is *negative sampling* Mikolov and Dean 2013; Mikolov et al. 2013, a variant of noise-contrastive estimation Gutmann and Hyvärinen 2012. Instead of considering the context's aggregated importance score, the learning algorithm with negative sampling processes every single co-occurrence of a target node and its context as a single training instance. A single co-occurrence is, e.g., a visit of a context node by a random walk. Given a pair of target and context nodes with their corresponding embeddings, the training algorithm maximizes the dot product between them. The algorithm selects several negative contexts and minimizes the similarity between their representations and the vector of the target node to avoid the degenerate solution:

$$\log \sigma(w_t^T w_c') + \sum_{j=1}^{k} \mathbb{E}_{j \sim P_n} \log \sigma(-w_t^T w_j')$$

the w_t is the embedding of the target node, w_c' of the context neighbor, w_j' of the negative sample and σ denotes the sigmoid activation. It is noteworthy that learning with negative sampling implicitly factorizes the co-occurrence matrix Levy and Goldberg 2014; Qiu et al. 2018.

Implicit or explicit factorization of the co-occurrence matrix results in two matrices. Let the matrix W_t contain the representations of the target nodes and the W_c the representations of context nodes. In the case of parameter sharing, if $W_c =: W_t$, the training goal is the similarity of the target node representation to the representation of its contexts. If there is an extra matrix for the representations of contexts $W_c \neq: W_t$, the relationship between two target nodes depends on the similarity of their contexts.

Besides the previously discussed shallow architectures, some methods use autoencoders with multiple layers with non-linearities Cao, Lu, and Xu 2016; Wang, Cui, and Zhu 2016. The autoencoder is trained to reconstruct the representation of the node co-occurrence vector.

Structural Roles Apart from the homophily assumption, another potentially powerful signal for the learning of node representations is the information about *roles* of the nodes in the graph. Intuitively, nodes having the same or similar topological or structural properties have similar roles and do not have to reside close to each other in the graph. In some applications, nodes modeling entities with the same function have a similar role in the network Kleinberg 1999; Scripps, Tan, and Esfahanian 2007; Luczkovich et al. 2003. Earlier work in the field mainly deals with the *role discovery* or the division of nodes into classes with the same roles. Over time, there were different suggestions for defining structural properties, and to what extent these properties of nodes have to be *equivalent* or *similar* for the nodes to belong to the same class. Previous works formulate different types of role equivalences, which differ in how strict the definition is Rossi and Ahmed 2014. For example, *structural equivalence* Lorrain and White 1971 requires nodes from the same class to be connected to precisely the same neighbors [17]. On the other hand, the *stochastically equivalent* Holland and Leinhardt 1981 nodes have a similar probability distribution of roles in the immediate neighborhood. Later approaches further relax the equivalence requirement and rely on the structural *similarity* Jin, Lee, and Hong 2011; Jin, Lee, and Li 2014 instead. Therefore they use hand-crafted structural feature vectors and assign nodes to the same class if they have similar features using, e.g., clustering or different types of matrix factorization. In more recent work the structural node descriptors were used for node classification and matching of graphs Bhattacharya and Getoor 2007; Henderson et al. 2011; Henderson et al. 2012; Gilpin, Eliassi-Rad, and Davidson 2013; Ribeiro, Saverese, and Figueiredo 2017; Donnat et al. 2018; Heimann et al. 2018. As we demonstrate in chapter 3 suitable node structural descriptors can also be used to characterize whole graphs. We aggregate role descriptors to the single graph representation and show that the resulting representation can be used for the graph classification.

The commonly used structural descriptors mostly describe a local topology of the node's local neighborhood. Many approaches use statistics about the node's degree and its local neighbors to create structural node embeddings Henderson et al. 2011; Henderson et al. 2012; Gilpin, Eliassi-Rad, and Davidson 2013; Ribeiro, Saverese, and Figueiredo 2017; Tu et al. 2018. Another exciting direction is the characteristic of the signal diffusion in the

[17] what contradicts the expectation that members of the same class do not have to be close to each other

local neighborhood. For example, the feature vector computed by *GraphWave* algorithm Donnat et al. 2018 describes the effect of filtering with the heat kernel in the spectral domain.

1.2.2 Semi-Supervised Learning on Graphs

Supervised learning describes a process of learning with a labeled dataset, where labels represent a target for the task of interest. The semi-supervised learning denotes the setting where unlabeled instances are additionally used in the learning process. Node classification often comes up together with semi-supervised learning. Some node classification approaches have initially been proposed for semi-supervised learning, where the node classification is executed on the k-nn similarity graph Joachims 2003. However, the general node classification task, where the only fraction of labels is known, can also be considered semi-supervised since unlabeled nodes are also used in the training process, e.g., to propagate label information of the neighbors.

Collective inference with relational classification One line of work for the semi-supervised node classification can be described by the combination of *relational classification* models with *collective inference* Macskassy and Provost 2007. In the first step, the relational classification model is trained based on the existing labels. Relational classification models predict class labels based on the labels in the node's immediate neighborhood. In the simplest case, under the homophily assumption, the most frequent label in the neighborhood is assigned to the node under consideration Macskassy and Provost 2003. More advanced approaches aggregate labels in the local neighborhood and use them as input to classification model Lu and Getoor 2003; Chakrabarti, Dom, and Indyk 1998. Alternatively, instead of learning a classification model, each class is represented by an average vector of label counts in the local neighborhood. Unlabeled nodes compare their label counts vector with the class representations, and the label of the class with the highest similarity is assigned to the node Perlich and Provost 2003; Macskassy and Provost 2007. Our approach presented in chapter 4 can also be considered as an instance of relational classification. In contrast to previous approaches, our method considers neighborhoods of various extensions and can also aggregate information from multiple neighborhoods. Therefore our approach is less affected by the problem when labeled nodes have no or very few neighborhood labels.

Collective inference describes the procedure when multiple instances are classified jointly, and predicted labels influence further predictions Jensen, Neville, and Gallagher 2004. When combined with relational classifiers, the relational classification model is first trained using known labels. In the second step, the same model is applied iteratively without retraining until the label predictions for the unlabeled nodes stabilizes. Approaches based on *Gibbs Sampling* Geman and Geman 1984 define random ordering of unlabeled nodes and sample a label from the predicted label distribution Sen et al. 2008. The neighbors utilize sampled labels as input to the classification model for the next predictions. *Iterative classification* Neville and Jensen 2000; Sen et al. 2008; Lu and Getoor 2003 works similarly

to the Gibbs Sampling; the main difference is that a class with the highest probability is taken directly instead of sampling. In contrast, the *Relaxation Labeling* Chakrabarti, Dom, and Indyk 1998 uses labels predicted in the previous iteration as input to the classifier, and labels predicted in the current iteration are not used until the next iteration. Note that collective inference methods require multiple hundred iterations and therefore are computationally expensive. Furthermore, the convergence is not always guaranteed; it was observed that they also do not always converge in the praxis Macskassy and Provost 2007. The collective inference approaches are necessary with relational classification models using only direct neighbors to make decisions for the nodes without labeled neighbors. In our approach in chapter 4, it is possible to parametrize the neighborhood extension. By selecting wide enough neighborhoods, it is possible to make sure that each node has labeled neighbors.

Label Propagation The label *spreading/propagation* (LP) Zhu and Ghahramani 2002; Zhu, Ghahramani, and Lafferty 2003; Zhou et al. 2004; Peel 2017; Wang, Tu, and Tsotsos 2013 and *loopy belief propagation* methods Pearl 1982; Yedidia, Freeman, and Weiss 2003; Koutra et al. 2011; Gatterbauer et al. 2015 can be formulated in the same message propagation framework. Each node maintains the 'belief' about the own label distribution and updates it iteratively based on the messages received from the neighbors. Iterations are executed until convergence is reached and the 'beliefs' aren't changing significantly.

LP methods mostly rely on the homophily assumption and, in the end, assign the most popular label in the local neighborhood. The main difference between different LP approaches is the assembling and normalization of a graph propagation matrix. In each iteration, the propagation matrix is multiplied with the label matrix. Therefore all messages are summed up, and each neighbor's contribution corresponds to the weight in the propagation matrix.

The *Belief Propagation* (BP) algorithm also uses message passing between neighbors and can be applied when the homophily assumption does not hold. It is also known as *sum-product* and it is a popular algorithm for the inference[18] in different types of probabilistic graphical models Kschischang, Frey, and Loeliger 2001 [19]. BP algorithms avoid the expensive enumeration [20] of all possible states by the utilization of conditional independence in the graphical model. Therefore the graphical model is often transformed into the factor graph Yedidia, Freeman, and Weiss 2003, which represents the joint distribution factorization. Nodes in the factor graph model variables and factor function on each edge define the joint probability of connected nodes' variables. In each iteration, all nodes send a message to the neighbors. The message indicates the 'belief' of the sender node about the receiver node's state. It depends on the sender's belief about its state and

[18]Inference is the process of computing marginal distribution for not observed random variables.

[19]Graphical model is a graph describing conditional dependence between random variables, where random variables are modeled by nodes and dependencies by edges.

[20]By the law of total probability the naive approach to compute a probability for a random variable to take on a particular value is, to sum up, the probabilities of all possible states, where the variable has this value.

factor function defined between the sender and receiver. Each receiver node multiplies all incoming messages with its own previous belief about its state to update it. BP computes exact marginal probabilities on the trees, and on the graphs only the approximation, the convergence is not guaranteed. The adaptation to the node classification scenario is trivial Sen et al. 2008. The BP algorithm expects a matrix with pairwise label affinities as input to model as an analog to the pairwise joint distribution. The message is built using local 'belief' about own label distribution and information about the labels' relationship. [21]

The main restriction of the label propagation approaches is that nodes have to be connected in the same connected component to exchange information. Nodes in different connected components cannot benefit from each other. Moreover, the methods are *transductive*; for the new nodes, the algorithms have to be executed from scratch.

Message-Passing Neural Networks Semi-supervised node classification on graphs with node attributes is popular task for MPNNs Hu et al. 2020. In non-attributed graphs, where only target labels of nodes are known, MPNN can utilize the learnable parameters as initial node embeddings. However, in this case, the node embedding receives gradient updates only if the node itself is labeled or its message directly or indirectly reaches the labeled node used to minimize the loss in the training process. Otherwise, if the node embedding does not affect some node's output representation used for loss computation, its initial embedding is not updated. Due to the oversmoothing [22] problem, typical MPNNs have a couple of layers. Therefore, each node takes only neighbors from the first few hops into consideration. Depending on the graph's structure, the number of labels, and how well the labeled nodes are distributed across the graph, some nodes can become no or not enough updates to learn something useful.

The MPNN models can also be applied in the *inductive* setting. The embedding of a newly connected node can be initialized using some aggregation of neighbor representations Hamilton, Ying, and Leskovec 2017.

[21] The label affinity vector is multiplied with the sender's label distribution to compute the belief about each receiver label. Therefore this step can be seen as partial marginalization.

[22] Oversmoothing describes the phenomenon when an increasing number of layers leads to similar output node representations for all nodes in the same connected component Li et al. 2019b

1.3 Graph Matching

The problem of Graph Matching (GM) or finding the correspondence between nodes of different graphs is important for spatial applications (chapter 6), various tasks in computer vision Guo et al. 2018; Iqbal, Milan, and Gall 2017, bioinformatics Singh, Xu, and Berger 2008, analysis of social networks Ahmad and Ali 2019 or transferring informations between different knowledge bases. For example, in computer vision applications, nodes often model detected objects, and the goal is to detect the same objects in different scenes. GM approaches use the fact that the objects' positions relative to each other are similar in different scenes and represent scenes by graphs, where edges model proximity between objects.

The procedure of graph matching usually involves two steps. First, an affinity matrix, which encodes similarities between elements of different graphs, needs to be defined. An affinity matrix encodes similarities between nodes, edges, or even higher-order structures such as hypergraphs Chertok and Keller 2010; Zass and Shashua 2008. Given the affinity matrix, the assignment procedure searches for the node-to-node correspondence, which maximizes the sum of similarities between matched elements, where the matching depends on the assignment of the nodes. For example the solution for the *Lawler Quadratic Assignment Problem* Lawler 1963 for two graphs $\mathcal{G}_1 = \{V_1, E_1\}$ and $\mathcal{G}_2 = \{V_2, E_2\}$ with $|\mathcal{V}_1| = n$, $|\mathcal{V}_2| = m$ and the affinity matrix $K \in \mathcal{R}^{mn \times mn}$ is a node correspondence matrix $P \in \{0,1\}^{m \times n}$ which maximizes the following quadratic form:

$$\max_P vector(P)^T K vector(P)$$

The matrix K, known as the affinity matrix of second-order, encodes node-to-node similarities by diagonal elements and affinities between edges in two graphs by the rest of the matrix. The assignment solution is often subject to one-to-(at most)one constraint.

In the following, we review two lines of work, where the optimal affinity matrix for the problem of graph matching is *learned* end-to-end. In the first setting, which we call *inductive graph matching*, we are given a dataset of graph pairs with the known node mappings between graphs, and the goal is to learn a model which can decide about node matchings for the new graph pairs which were not seen by the model in the training procedure. In another line of work known as *Entity Alignment* the aim is to learn a model that can match nodes of two given *Knowledge Graphs* (KGs) in the transductive semi-supervised setting. Given the fraction of mappings between two KGs, the goal is to imply the rest of the matchings.

Note that there is a family of approaches for graph matching dealing with the node matching assignment based on a predefined affinity matrix. It is still an active area of research Gold and Rangarajan 1996; Leordeanu and Hebert 2005; Cho, Lee, and Lee 2010; Bernard, Theobalt, and Moeller 2018; Yu et al. 2018; Wang et al. 2020, and these methods can be used complementary to the approaches learning an affinity matrix at the inference time.

1.3.1 Inductive Graph Matching

Feature learning The starting point for the learning of an optimal affinity matrix is the definition of the node and edge features. Earlier approaches for computer vision tasks use hand-crafted features Caetano et al. 2009; Leordeanu, Sukthankar, and Hebert 2012; Cho, Alahari, and Ponce 2013 and learn the importance of each feature for the optimal affinity or distance score. More recent works apply neural networks for the learning of optimal feature representations. Most works follow the Siamese architecture Bromley et al. 1994, where an identical network is used to extract features for nodes of both graphs. For example Convolutional Neural Networks (CNN)s are employed for the feature extraction Zanfir and Sminchisescu 2018; Wang, Yan, and Yang 2019; Yu et al. 2019; Fey et al. 2020 in computer vision applications. When visual features are not available, coordinates of the corresponding nodes on the image Zhang and Lee 2019 are used as input to the MLP. In the case of not-attributed graphs, the degree of the nodes or sum of degrees in its k-hop degree neighborhood were successfully used as input to the neural network Nowak et al. 2017; Nowak et al. 2018.

Models One of the first end-to-end learning approaches Zanfir and Sminchisescu 2018 decomposes the affinity matrix of second-order analogously to Zhou and Torre 2012 into a product of node and edge representations, which are computed by CNN. Later approaches employ Message Passing Networks (MPNN) on the top of feature extractor Nowak et al. 2018; Wang, Yan, and Yang 2019; Yu et al. 2019; Fey et al. 2020. The basic architecture, which was first proposed in Nowak et al. 2018, is similar in most methods using MPNNs. The Siamese MPNN outputs the nodes' representations, and the dot product of node representations computes the affinity matrix. The similarity matrix is often normalized by (few iterations of) Sinkhorn procedure Sinkhorn 1964; Knight 2008, which makes the affinity matrix double stochastic.

Graph Structure The graphs for the matchings are usually provided by the application. For most computer vision tasks, the graph provides information about the nodes' spatial positions relative to each other. Therefore, the representation of the node outputted by MPNN aggregates the node features with the features of other nodes in its spatial proximity. An unusual alternative is proposed in Wang, Yan, and Yang 2019, where in each layer, a similarity matrix between node representations of both graphs, is used as adjacency. Therefore, each node receives information from its most similar nodes in another graph in each layer. Later the architecture was extended to the alternating application of similarity and spatial adjacency matrices Yu et al. 2019.

Loss Zanfir and Sminchisescu 2018 proposes *displacement loss*, which uses the model to map each node's position from one of the graphs to the domain of other graphs and minimizes the distance to the ground truth matching. The other domain position is computed as the weighted mean of nodes in other graphs, where matching probabilities in the affinity matrix computed by the model are used as weights. Therefore, the displacement loss can

only be used in the scenarios where nodes have spatial coordinates, or there is a meaningful distance function for the nodes of the same graphs. Other works Nowak et al. 2018; Wang, Yan, and Yang 2019; Yu et al. 2019; Fey et al. 2020 use more general applicable binary cross-entropy, which is applied element-wise on the normalized affinity matrix. Yu et al. 2019 additionally proposes the approach called *hungarian attention*, where the loss is applied on the part of the affinity matrix. In each forward path, the *hungarian algorithm* Kuhn 1955 computes matchings based on the affinity matrix. The wrong matches assigned by the Hungarian algorithm are used as negative instances for the loss computation, while all true matchings are still used as positives.

1.3.2 Entity Alignment

Knowledge Graphs (KG)s have their origin in the Semantic Web, and the main goal is to organize information so that it can be read and interpreted to some extent by machines Lenat and Feigenbaum 1992. This capability enables the utilization of information from KGs by different computer-supported applications requiring access to knowledge. KGs are used to retrieve information about entities in search applications Singhal 2012 or answer natural language queries Zhang et al. 2017; Saxena, Tripathi, and Talukdar 2020, in various biological applications such as drug discovery Mohamed, Nounu, and Nováček 2019 or used as a knowledge base for the conversational bots Tian et al. 2020.

A KG is a knowledge base where information is structured in the form of triple facts. The core of each KG is formed by real-world *entities*, which are real-world animate and inanimate objects described by the facts. Each fact is a single statement about some entity expressed in the form of triple *subject-predicate-object*. A subject is always an entity, and if the predicate is a *relation*, the fact describes a relationship between two entities. There are also *attribute* predicates, and in this case, an object in the fact is a value of primitive data type Lin, Liu, and Sun 2016. Each entity and relation in KG is uniquely identifiable by a URI, and therefore the knowledge base in the form of triples can be represented as a heterogeneous graph. Each entity in the graph is modeled by a node, attributes by node attributes, and for each relation fact, there is an edge between two nodes with the corresponding type. The resulting relation graph can be represented by a three-dimensional tensor of dimensionality $r \times \mathrm{N} \times \mathrm{N}$, where r is number of relation types and N number of entities.

The completeness of information in KG is an essential factor for applications relying on it. Today KGs are employed by the largest technology companies Noy et al. 2019 and already contain hundreds of billions of facts about hundreds of billions of entities extracted from various data sources Dong et al. 2014. There are different approaches to complete KG with new facts. *Link Prediction*, also known as *reasoning*, is the process of inferring new facts based on the previous information in Knowledge Graph. Representation Learning approaches for Link Prediction Nickel et al. 2015 learn embeddings of entities and relations, and the probability of the new triple can be estimated using the combination of corresponding embeddings. Another family of representation learning approaches, known as Entity Alignment (EA), enables information exchange between different KG by matching

their entities. The basic assumption behind all Entity Alignment methods is that the significant fraction of facts in both KG is the same. Therefore, the high-level idea is to use facts in each KG to organize representations of entities and optionally relations in some vector space. An important feature of different approaches is the information utilized by the model for the EA task. All approaches use the KG structure and differ in how and whether at all other knowledge is used. The main inconsistencies can be observed for the usage of the following information:

- Attributes: Attributes are often very sparse, and usually, the utilization of the attributes only slightly improves the matching result, and therefore, they are not used by the majority of the methods.
- Entity Names: Although entity names can also be considered attributes, they often become different treatment. Names provide a powerful signal in all used benchmarks (section 10), and often, methods exploiting it are compared to approaches that do not use entities' names for the matching.
- Types of relation: Different edge types are a distinctive feature of KGs. However, many methods aggregate them and create a homogeneous graph.

If KGs share similar information, the representation of each of two aligned entities is positioned similarly relative to other elements from the own knowledge base in the vector space. Since the training procedure forces aligned entities from the training set to end up close to each other in the common vector space, similar relationships lead to close proximity for the rest of the aligned entities. In the following, we discuss two types of models for the EA task.

Link Prediction Models One family of approaches for EA makes use of Link Prediction (LP) model to create meaningful embeddings of entities and relations Hao et al. 2016; Sun, Hu, and Li 2017; Chen et al. 2017; Sun et al. 2018; Pei et al. 2019; Qu, Tang, and Bengio 2019; Chen et al. 2018; Lin et al. 2019; Trisedya, Qi, and Zhang 2019; Shi and Xiao 2019; Zhang et al. 2019. While optimizing for the LP task, methods have an additional loss, which forces aligned entities to be close to each other in some shared vector space. Most of these methods use *TransE* model Bordes et al. 2013, which models a relation as translation in embedding space and minimizes the distance between the representation of the tail entity and sum of head and relation vectors. There is also a method using a combination of more powerful models instead of TransE, and evaluation shows that it leads to better performance Shi and Xiao 2019. Likely, the translation model cannot handle all relation patterns occurring in typical KG. An example of such a pattern is when the same entity is related to multiple entities by the same relation. In this case, the method forces these entities to have an equal representation. A similar effect can be observed with several different relations between the same pair of entities.

The typical procedure to represent elements of both graphs in the same vector space is to define them in two distinct spaces and to learn an additional linear transformation to map

between them. Besides, more recent approaches utilize the adversarial learning framework Goodfellow et al. 2014 to improve the overlap of the original and translated representations. Therefore, in combination with an extra discriminator model, an additional loss function provides feedback for the function responsible for the mapping between different embedding spaces. For instance Lin et al. 2019 aims to make original and mapped embeddings, and Qu, Tang, and Bengio 2019 triples formed by these embeddings non-distinguishable. There are also works adapting cycle loss Zhu et al. 2017 for the task of EA Pei et al. 2019; Lin et al. 2019 analogously to its usage for the alignment of word embeddings from different vocabularies Conneau et al. 2017. After mapping to another space, an embedding is mapped back to the original space, and the loss minimizes the distance to the original representation. There are also other possibilities to represent embedding in the joint space, e.g., to learn a translation vector instead of linear transformation Chen et al. 2017 or merge both graphs based on the provided alignments and to learn embeddings for the merged graph Sun, Hu, and Li 2017.

To make use of attributes, LP approaches utilize additional models representing entities based on attributes or names Zhang et al. 2019; Chen et al. 2018. These models are trained with the own loss function and the final affinity matrix is a combination of similarities produced by LP and attribute models.

Message Passing Networks Another line of work utilizes Message Passing Networks (MPNN) to summarize the relevant facts in representations of the entities Wang et al. 2018; Sun et al. 2019; Zhu et al. 2019; Cao et al. 2019; Li et al. 2019a; Wu et al. 2019b; Wu et al. 2019c; Mao et al. 2020; Xu et al. 2019; Fey et al. 2020. MPNNs are used in Siamese architecture, where the same model is applied to both KGs and outputs representations of the entities. Models are trained end-to-end and learn to maximize the similarity of representations of aligned entities in the common vector space.

The vital distinction among MPNN based methods is whether information about entity names is taken into account. If the methods do not consider entities' names, the initial embedding of the node representing an entity is randomly initialized and is updated in the course of training Wang et al. 2018; Sun et al. 2019; Zhu et al. 2019; Cao et al. 2019; Mao et al. 2020; Li et al. 2019a. In contrast, models employing entity names utilize word embeddings for the initial representation of the entities Xu et al. 2019; Fey et al. 2020; Wu et al. 2019b; Wu et al. 2019c. These initial representations are either used to initialize trainable embeddings or as fixed node features and stay unchanged during the training. Since KGs for the matching are defined in different languages in many benchmark datasets, models employ cross-lingual embeddings Conneau et al. 2017; Feng et al. 2020 or translate names from one of the KGs to another language with the external translation tool. When using entity names, the matching zero-shot performance is better than most methods' performance using random initialization. The zero-shot performance is computed using the affinity matrix representing the similarity of the entity names. The entity names are represented by word embeddings, which were pre-trained using some independent tasks. Therefore the distinction between methods utilizing random initialization and methods

using entity names is essential for analyzing progress in the field.

Most of the MPNN based approaches do not take information about different types of relations into account. KG tensor is often aggregated to the binary adjacency matrix indicating, whether there is at least one connection between two entities. Other approaches use relations to define edge weights by some heuristic, which takes, e.g., the number of relations between entities and their popularity in the graph into account Wang et al. 2018. Intuitively, methods able to learn analogy of relations should enable more fine-granular comparison and better matching. There are already a few MPNN based approaches considering different relations in the message passing process. In Mao et al. 2020, authors propose to concatenate the mean of representations of adjoining relations to the entity embedding. An alternative approach is proposed in Wu et al. 2019b, where nodes model relations and message passing is additionally performed on the resulting dual graph.

1.4 Argument Mining

Argumentation is the process of identification and comparison of reasons supporting or opposing some standpoint. The main goal is to provide support by forming an opinion about a controversial topic or resolving differences in viewpoints. Various aspects of argumentation are researched by different disciplines, such as logic, philosophy, psychology, cognitive science, or linguistics. *Argument Mining* (AM), also known as *Computational Argumentation* is a subfield of computer science, which studies the automatic detection and retrieval of relevant argumentative structures. In contrast to *sentiment analysis* or *opinion mining*, which aim to recognize the view of a writer on some topic, AM's goal is to determine why someone may come up with an opinion. The main tasks of AM are the detection of argumentative components in unstructured text, estimation of their relevance, identification of relationships between components of a single argument or how different arguments interact with each other Cabrio and Villata 2018; Lawrence and Reed 2020. These tasks are often directly formulated as Machine Learning (ML) problems, or ML approaches are employed in intermediate steps. ML methods can also be combined with reasoning approaches for more advanced solutions Galassi et al. 2020.

As AM is a relatively new field, novel argument structures or AM applications in new domains are actively investigated. However, there is no uniform view on the definition of the argument, and the current state in argumentation theory is that there are competing views on different aspects of argumentation Van Eemeren, Grootendorst, and Kruiger 2019; Habernal and Gurevych 2017. Depending on the domain and task requirements, there are various possibilities to model the argumentative structure. From the application perspective AM approaches can be categorized based on the types of texts they are applied to. One speaks about *Discourse Level* AM, when the source documents are inherently argumentative and common argumentative structure can be assumed. Examples include legal documents Palau and Moens 2009, persuasive essays Stab and Gurevych 2017; Eger, Daxenberger, and Gurevych 2017; Nguyen and Litman 2018, political debates Naderi and Hirst 2015; Lippi and Torroni 2016a or research articles Teufel, Siddharthan, and Batchelor 2009. On the other hand, *Information-Seeking* AM aims to analyze arguments in texts, which are not necessarily argumentative by nature. Examples for that are Wikipedia articles Levy et al. 2014; Rinott et al. 2015, internet blogs, news, forum posts Habernal, Eckle-Kohler, and Gurevych 2014; Habernal and Gurevych 2017; Ein-Dor et al. 2020 or any types of texts found online Stab, Miller, and Gurevych 2018. Furthermore, in chapter 12, we demonstrate AM's usefulness for analyzing peer-reviews for scientific publications. When introducing a new task in AM, the usual approach is to perform an annotation study with several annotators and analyze the meaningfulness of the new task based on the agreement among annotators. When annotation agreement is good enough, the proposed task can be automated, e.g., by applying a ML model. Although a lot of effort is put into clarification and formalization of definitions, argumentativeness remains a subjective matter, and AM tasks are characterized by lower agreement scores than in other NLP areas Habernal and Gurevych 2017.

Argument Detection A widely accepted view, known as *structured* Lippi and Torroni 2016b or *micro-level* Bentahar, Moulin, and Bélanger 2010 argumentation in the argumentation theory, is that an argument comprises multiple components which interact with each other. Consequently, the parts of the same argument are scattered across the text, and an AM approach is responsible for identifying argumentative components in the text, their classification, and establishing relationships. The argument structure is formalized in *argument scheme*, which defines argumentative components, functions, and interactions between them. For instance, the well known Toulmin scheme Toulmin 1958 describes an argument as subsumption of argumentative components of six types. However, argumentation structure in many types of documents does not explicitly follow Toulmin model Newman and Marshall 1991. Therefore, various schemas were proposed in the literature Lawrence and Reed 2020, whereas, over time, it has become apparent that argumentation across multiple text types has a simple structure. As a result, a schema comprising *claim* and *evidence* components became popular in the recent related work, e.g. Levy et al. 2014; Habernal, Eckle-Kohler, and Gurevych 2014; Rinott et al. 2015; Habernal, Eckle-Kohler, and Gurevych 2014; Ein-Dor et al. 2020. Claim, in some works also called a *motion*, is a statement about a topic with a clear stance an argument wants to prove, e.g. *"We should ban the Nuclear Power"*. Evidence, also known as *premise*, provides explicit justification for the supporting or attacking the claim, e.g. *"Nuclear power plants produce no greenhouse gas emissions during operation"*, or *"Nuclear power plant accidents may have horrific consequences"*. Moreover, works aiming to develop AM models, which generalize to various types of documents, further reduced the schema to the single evidence components in combination with predefined topic Stab, Miller, and Gurevych 2018; Trautmann et al. 2020.

Argument Retrieval AM enables the extraction of arguments from various sources and an unlimited number of documents. There are many possible applications, which can make use of the vast number of potential arguments. To make the arguments accessible for the final users, some sort of retrieval system is necessary. Dependent on the application, various retrieval tasks are conceivable. Currently available argument search engines expect a topic as a query and perform a search based on the similarity of query text to the argument's text. For instance *www.args.me* searches through arguments already pre-extracted from debate portals and utilizes BM25 similarity for retrieval Wachsmuth et al. 2017. In contrast, *www.argumentsearch.com* analyzes retrieved documents for the argumentative structures in an online fashion but still performs classic document search based on the text similarity Stab et al. 2018. As text similarity-based approach can potentially miss relevant arguments, alternative retrieval approaches Potthast et al. 2019 or more advanced retrieval tasks are investigated in the related work. For instance retrieving evidence for the query in form of a claim allows more fine-granular search Dumani and Schenkel 2019; Dumani, Neumann, and Schenkel 2020; chapter 13.

There are various factors, which can be relevant for the ranking of the retrieved results. The diversity of retrieved arguments plays an important role for various applications Du-

mani, Neumann, and Schenkel 2020; chapter 13. Another relevant factor is the argument convincingness. One possibility to estimate the power of arguments is a manual comparison with the counterarguments. There are already some works addressing the problem of counterargument search Carstens and Toni 2015; Cocarascu and Toni 2017; Habernal et al. 2018; Wachsmuth, Syed, and Stein 2018; Orbach et al. 2020. However, if the search algorithms have to take argument strength into account, some sort of automation is necessary. It is an actively researched topic nowadays, which is addressed from different directions. There are works investigating quality features of arguments Lauscher et al. 2020; Wachsmuth and Werner 2020 or relying on machine learning to compare arguments with each other Habernal and Gurevych 2016; Simpson and Gurevych 2018; Gleize et al. 2019 or to assign a global quality score Swanson, Ecker, and Walker 2015; Toledo et al. 2019; Gretz et al. 2020.

1.5 Overview of the Thesis

In this thesis, we make several contributions in the area of Representation Learning on Relational Data. Therefore we start with the prediction problems on graphs and introduce several unsupervised and semi-supervised learning methods. Afterward, we move our focus towards the problem of graph matching. We contribute to the modeling part and the label acquisition, evaluation, and benchmarking procedures in this field. Next, we explore the subfield of Natural Language Processing known as Argument Mining and introduce new approaches that use relationship information. Finally, we present our scalable approach for graph generation, where the graph exhibits the latent cluster structure in the dataset. To this end, the contributions presented in the thesis address the following research questions:

Research Question 1: *What is the best way to learn meaningful node embeddings in an unsupervised fashion, and for what can these embeddings be useful?*

A strong inductive bias about the meaning of relationships in the graph is essential for unsupervised learning on graphs. The *homophily* based approaches for the unsupervised learning of node embeddings assume that each node is similar to its close neighbors and therefore resulting representations depend on the local neighborhood of the node provided by the method. However, as we demonstrate in chapter 2 on the example of popular benchmark datasets, community profiles of the real-life graphs do not exhibit cluster structure, and therefore the local structure of the neighborhood is challenging to grasp. Therefore, we introduce a novel approach for the unsupervised learning of node embeddings based on the homophily assumption called *LASAGNE*, which retrieves only the most relevant neighbors for each node. We demonstrate the advantages of our method empirically and show that it is also flexible regarding the local neighborhood's extension.

Furthermore in chapter 3 we introduce novel *role-based* node feature descriptors. The distinguishing feature of the role-based node representations is that they describe the node neighborhood's topological properties. Therefore, two nodes with similar roles do not necessarily share the same neighbors and may even reside in different graphs. The node role feature descriptors in our approach are computationally inexpensive and, at the same time, achieve better performance than previous state-of-the-art approaches on typical benchmarks. Additionally, we demonstrate that our feature descriptors can be successfully utilized to define graph representations for graph level prediction tasks.

Research Question 2: *Can we define a model that is able to make predictions about target variables of nodes only based on relationship information in the graph without making rigid assumptions about the meaning of relationships?*

While unsupervised learning approaches require some additional task to learn something useful, semi-supervised learning methods can exploit the target task's signal. In chapter 4, we address the problem of semi-supervised node classification on graphs, where nodes

have only target attributes. Our approach, called *ADA-LLD*, defines a representation for each node that describes the distribution of labels in the node's local neighborhood. The end-to-end training procedure learns a correlation between label distribution in the neighborhood and the node's label instead of making assumptions about the relationship type. Our method's unique feature is that it can vary a neighborhood's locality or even combine neighborhoods of different localities in a single model. We show that our approach outperforms standard methods for the node classifications by a large margin.

Moreover, in chapter 5, we introduce a semi-supervised approach to the problem of spatial interpolation. Standard approaches for this problem rely on some manually defined distance function, which integrates information about relations. Our approach is more generally applicable since it can learn an optimal distance function based on the available relationship information in an end-to-end fashion.

Research Question 3: *Can neural network-based graph matching techniques be of help if graphs are different and have only partial overlap?*

Graph matching or alignment methods aim to exploit the relationships in graphs to allow better matching of entities modeled by the nodes. The underlying assumption is that graphs are created by a similar process, and therefore there are analogous relationships between nodes. However, real-life data is often noisy and incomplete. As we demonstrate in chapter 6 on the example of maps, graphs we have to deal with can differ significantly. To address this problem, we introduce a novel method for the learning of optimal graph matching called *GrAN*. Our approach's main idea is to learn to ignore the effect of nodes appearing only in one of the graphs.

Research Question 4: *What do we need for the convincing evaluation and benchmarking of different Entity Alignment approaches?*

The empirical evaluation is one of the crucial elements for the assessment of scientific progress. A necessary requirement for a convincing and a fair evaluation is an informative evaluation score, a meaningful benchmark task, and fair experimental setting. In chapter 8, we carefully examine the typical metrics reported in the fields of Entity Alignment and Link Prediction in Knowledge Graphs and identify several shortcomings. Therefore, we introduce a novel evaluation score and demonstrate empirically that its utilization leads to more infallible and interpretable evaluation. Furthermore, in chapter 10, we analyze the typical evaluation setting of Entity Alignment approaches and observe that reported results are not reliable, and a comparison between different approaches is not possible. Therefore, we fix the experimental setting and provide an extensive comparison of state-of-the-art methods and make several interesting findings. Moreover, in chapter 9, we analyze the basic architecture for the Entity Alignment task and report several not self-evident findings regarding the relevance of the various architecture choices for the final performance of the model.

Research Question 5: *Can the label acquisition process for Entity Alignment be designed more efficiently than when the labeling pairs are selected randomly?*

The current Entity Alignment approaches require thousands of labeled pairs, which are expensive to acquire in a real-life situation. In chapter 7, we motivate the new problem of label acquisition for the Entity Alignment task, where the goal is to minimize the labeling costs. We introduce a labeling setting for the problem and propose and evaluate various heuristics with different assumptions. Our evaluation confirms that using suitable heuristics, we can significantly reduce labeling costs.

Research Question 6: *Can we improve argument mining models by taking into account the information about relationships between arguments?*

In chapter 11, we address the problem of *argument identification*, where the goal is to estimate the argumentativeness of text pieces extracted from an unstructured corpus. We argue that a topic of the potential argument provides relevant context for the task and, therefore, should be taken into consideration. We investigate different architecture choices that enable taking the relationship between topic and argument into account and show that our transformer based model outperforms previous state-of-the-art approaches. Furthermore, in chapter 12, we propose to analyze the decision-making process in scientific publishing through the lens of argumentation and formalize it as an argumentative process. We analyze the generalization of models trained on pre-existing datasets and motivate the necessity for a new corpus, which we collect, annotate, and make available for the community. Using our method introduced in chapter 11, we show that arguments in peer reviews can be detected automatically with nearly human performance and that these arguments play a crucial role in the decisions about paper acceptance.

After arguments are extracted, they have to be made available for the end-users. In chapter 13, we focus on the problem of argument search or argument retrieval, where the goal is to retrieve relevant and non-redundant arguments for the query formulated by end-users from the database with arguments. The previous state-of-the-art method already recognized that text similarity between query and arguments does not suffice to find all relevant arguments and therefore searches for the similar queries in the database and retrieves arguments manually assigned to them. We introduce a novel approach, where we use a model to estimate the relevance of the arguments directly, without a detour over similar queries. We demonstrate that the appropriately designed model can capture the relevance relationship between query and argument. Furthermore, to ensure that retrieved arguments do not repeat semantically, we propose to describe each argument by its relationship to possible queries. The assumption behind this approach is that arguments with similar meanings have comparable relationships. Our empirical evaluation demonstrates the validity of this assumption. It shows that our method improves previous state-of-the-art results while using less information, which makes the approach more flexible in applying.

Research Question 6: *Can we develop an inductive model that learns to generate a graph with a community structure reflecting latent cluster structure in a dataset without explicit relationship information?*

In chapter 14, we introduce a novel scalable algorithm for subspace clustering. Our approach learns to recognize different subspaces by building a graph, connecting instances where objects in the same subspaces build graph communities. Our approach's distinguishing feature is that it is inductive and can assign instances not used in training to the clusters. In contrast, related approaches require retraining of the complete model. Furthermore, our method supports batch-wise training and can be applied for the clustering of large datasets.

Chapter 2

LASAGNE: Locality And Structure Aware Graph Node Embedding

The chapter includes following publication (omitted due to copyright reasons):

Evgeniy Faerman, Felix Borutta, Kimon Fountoulakis, and Michael W Mahoney. "Lasagne: Locality and structure aware graph node embedding". In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. **Best Student Paper Award**. IEEE. 2018, pp. 246–253. DOI: 10.1109/WI.2018.00-83

Chapter 3

Structural Graph Representations based on Multiscale Local Network Topologies

The chapter includes following publication (omitted due to copyright reasons):

Felix Borutta, Julian Busch*, Evgeniy Faerman*, Adina Klink, and Matthias Schubert. "Structural Graph Representations based on Multiscale Local Network Topologies". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. *equal contribution. IEEE. 2019, pp. 91–98. DOI: 10.1145/3350546.3352505

Chapter 4

Ada-LLD: Adaptive Node Similarity for Node Classification Using Multi-Scale Local Label Distributions

The chapter includes following publication (omitted due to copyright reasons):

Evgeniy Faerman, Felix Borutta, Julian Busch, and Matthias Schubert. "Ada-LLD: Adaptive Node Similarity for Node Classification Using Multi-Scale Local Label Distributions". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020). **Best Student Paper Award**, pp. 25–32

Chapter 5

Spatial Interpolation with Message Passing Framework

The chapter includes following publication (omitted due to copyright reasons):

Evgeniy Faerman, Manuell Rogalla, Niklas Strauß, Adrian Krüger, Benedict Blümel, Max Berrendorf, Michael Fromm, and Matthias Schubert. "Spatial Interpolation with Message Passing Framework". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2019, pp. 135–141. DOI: 10.1109/ICDMW.2019.00030

Chapter 6

Graph Alignment Networks with Node Matching Scores

The chapter includes following publication:

Evgeniy Faerman, Otto Voggenreiter, Felix Borutta, Tobias Emrich, Max Berrendorf, and Matthias Schubert. "Graph Alignment Networks with Node Matching Scores". In: *Graph Representation Learning NeurIPS 2019 Workshop*. 2019

Graph Alignment Networks with Node Matching Scores

Evgeniy Faerman[1] **Otto Voggenreiter** [2] **Felix Borutta**[1]
Tobias Emrich[2] **Max Berrendorf**[1] **Matthias Schubert**[1]
[1] *Ludwig-Maximilians-Universität München* Munich, Germany
{faerman, borutta, berrendorf, schubert}@dbs.ifi.lmu.de
[2] Harman International, Germany
{otto.voggenreiter, tobias.emrich}@harman.com

Abstract

In this work we address the problem of graph node alignment at the example of Map Fusion (MF). Given two partly overlapping road networks, the goal is to match nodes that represent the same locations in both networks. For this task we propose a new model based on Graph Neural Networks (GNN). Existing GNN approaches, which have recently been successfully applied on various tasks for graph based data, show poor performance for the MF task. We hypothesize that this is mainly caused by graph regions from the non-overlapping areas, as information from those areas negatively affect the learned node representations. Therefore, our model has an additional inductive bias and learns to ignore effects of nodes that do not have a matching in the other graph. Our new model can easily be extended to other graph alignment problems, e.g., for calculating graph similarities, or for the alignment of entities in knowledge graphs, as well.

1 Introduction

In recent years the high relevance of graph-structured data has generally been accompanied by an increased demand for algorithms that are able to take advantage from the rich body of relational information encoded in graphs. However, in many domains, the entire knowledge base of a certain domain is spread across multiple data sources, e.g., geo-spatial information are spread across various map providing services, social information are spread across multiple social networks, or, somewhat more general, knowledge bases are distributed across various knowledge graph databases.

In this work, we tackle the problem of knowledge fusion by developing a graph neural network model that is able to fuse graph-structured data by learning node matchings. Given two partly overlapping graphs, our approach aims at identifying nodes from both graphs that match to each other. The ultimate goal is to align the graphs such that the information contained in both graphs can be fused properly. In general, the node matching problem is of high relevance in many applications including the fusion of road networks, also called map fusion [1], the matching of knowledge graphs [2, 3, 4, 5, 6, 7, 8, 9, 10], the alignment of social networks [11], or to enable efficient graph comparison [12, 13]. Traditional methods addressing the graph alignment problem typically rely on calculating distances between manually engineered node and edge features. The distances are used to generate pairs of nodes that serve as matching candidates and the set of candidates is subsequently optimized by incorporating the nodes' neighborhood information [1, 14]. However, those approaches require non-trivial, manual parameter tuning, do not generalize well to unseen data and suffer from high complexities which makes them impractical. More recent works have applied graph neural networks (GNNs) which have proven to be particularly suitable for the graph alignment task [8, 9, 10, 12, 13]. The common approach is to aggregate nodes from the local neighborhood into a target node's representation and subsequently compare the resulting representations with each other. More advanced approaches even additionally aggregate nodes from the other graph into the node representations [12, 9]. Either way,

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

the aggregation of information, i.e., the kind of information as well as the locality of the area from which the information is aggregated, is the critical point to get useful embeddings. This holds for the graph alignment task in particular, as information gathered from local node neighborhoods generally tends to become less useful the more the two graphs that shall be aligned differ in terms of structural properties. To overcome the issue of aggregating irrelevant or even misleading information, state-of-the-art approaches use different types of attention mechanisms when aggregating node neighborhoods. The general idea behind using attention is to determine the importance of a certain entity (e.g., a node in the neighborhood) based on an object's own representation (e.g., the representation of the target node) and the current representation of the entity. The higher the importance, the more influence should the corresponding entity have on the object's own representation. However, the attention based on the structure of the own graph is of limited usefulness for the graph alignment task. This problem has recently been addressed, e.g., by additionally aggregating information from the whole counterpart graph into the representation of each node [12]. We argue that the most important information about a node's neighbors is whether they have good alignments in the counterpart graph. Therefore, the Graph Alignment Network (GrAN) presented in this work uses an importance mechanism that, in contrast to previous works, aims at putting special emphasis on nodes that have a good match in the counterpart graph. By doing this, our model effectively introduces an additional inductive bias, i.e., the assumption that neighboring nodes which are likely to be part of the overlapping area of the two graphs are particularly useful for a target node's representation.

After presenting our GrAN model in Section 2, we present preliminary results of our evaluation in Section 3. Precisely, we compare our model against state-of-the-art GNN approaches on map fusion tasks. These tasks turned out to be particularly challenging due to presence of geo-spatial coordinates as they form a natural and very strong baseline. However, it is noteworthy that our model can also be applied to other tasks including knowledge graph matching and the determination of graph similarities.

2 Graph Alignment Networks with Node Matching Scores

Let $G = (V, E, X, P)$ and $G' = (V', E', X', P')$ denote two graphs with V, V' denoting the sets of nodes, E, E' being the sets of edges and X, X', and P, P' being the node and edge attributes, respectively. For the sake of simplicity we assume that both graphs are undirected. Given this setup, we aim at learning a function $F : G \times G' \rightarrow H \in \mathbb{R}^{|V| \times h}$ which takes two graphs as input, applies multiple message passing operations on them and finally retrieves latent vector representations for the nodes of the first graph that was fed as input into F. For the GrAN model, we use a Siamese architecture that allows to apply the same function F to both graphs such that the model's final output are two node embedding matrices $H = F(G, G')$ and $H' = F(G', G)$, respectively. Given the node embeddings, we subsequently align two nodes if their vector representations are considered similar with respect to some similarity function sim and a predefined similarity threshold τ, i.e.,

$$\hat{y}(v_i, v'_j) = \begin{cases} 1 & if \quad sim(h_i, h_j) > \tau \\ 0 & else. \end{cases} , \tag{1}$$

with $v_i \in V$, $v'_j \in V'$, h_i denoting the row vector from H that corresponds to the embedding of node v_i and h_j being the row vector from H' that corresponds to the embedding vector of node v'_j. $\hat{y}(v_i, v'_j) = 1$ indicates that the nodes v_i and v'_j are aligned.

Graph Alignment Network. Graph Neural Networks [15, 16] compute node representations by propagating information between vertices and aggregating them iteratively. In each layer, a message is formed from each source node and passed to its neighbors. Incoming messages for each target node are aggregated and flow into the target node's output representation. Stacking multiple propagation layers allows propagating information over multiple hops. Given a graph $G = (V, E, X, P)$ we define a single message passing propagation step as follows:

$$m_{j \rightarrow i} = M^l(h^l_j, p_{ji}),\ with\ p_{ji} \in P, \tag{2}$$

$$M^l(h^l_j, e_{j,i}) = f_{message}(h^l_j) + LSTM(p_{ji}), \tag{3}$$

$$h^{l+1}_i = [f_{node}(h^l_i), \sum_{j,(j,i) \in E} \alpha^l_{j \rightarrow i} m_{j \rightarrow i}] \tag{4}$$

where, l stands for the current layer, M is a message function, f_{node} and $f_{message}$ are small neural networks and $[\cdot]$ denotes the concatenation operation. Also, note that for the Map Fusion task, the

edge features $p_{ji} \in P$ are sequences of coordinates that represent the road segment. Therefore, we process edge features with a recurrent LSTM network and add the resulting vector to the transformed node representations. However, the inclusion of edge features is generally optional. The weight $\alpha^l_{j\to i}$ determines the importance of the message emitted by node j. In fact, the α weights realize the additional inductive bias of our model, i.e., that the effect that a message has on other nodes is determined by the maximal matching similarity of the source node to the other nodes in other graph. Intuitively, assume that node $v \in V$ has two neighbors $u, w \in V$ with u being aligned to node $u' \in V'$ while node w has no matching node in V'. In this case, our goal is to put high emphasis on u and low emphasis on w when aggregating incoming messages of v. The intuition is that a node $v' \in V'$ who has u' as a neighbor also receives highly weighted information from u' which is assumed to be similar to the information from u. Therefore, in each layer, our approach first determines the best matching for each node with respect to the similarity function sim, and all outgoing messages are weighted accordingly. More formally, we define the message weight $\alpha^l_{j\to i}$ as follows:

$$\alpha^l_{j\to i} = \frac{exp(I(h^l_j, H'^l))}{\sum_{\hat{j}} exp(I(h^l_{\hat{j}}, H'^l))},\ with\ (\hat{j}, i) \in E,\ and \tag{5}$$

$$I(h^l_j, H'^l) = \max_k sim'(f_{match}(h^l_j), f_{match}(h'^l_k)),\ with\ h'^l_k \in H'^l. \tag{6}$$

Since it is not possible to backpropagate through the max operation, we apply the gumbel softmax trick [17, 18]. For the functions $f_{message}$, f_{node} and f_{match}, we use Multilayer Perceptrons and add an additional inductive bias by sharing parameters between these three functions. The inverse Euclidean function is used as similarity function $sim'(a, b) = \frac{1}{d_{euclidian}(a,b)+1}$.

Learning. We use the contrastive loss [19] with positive ϵ^+ and negative ϵ^- margins, i.e.,

$$L = y_{i,j} \cdot max(0, d(h_i, h_j) - \epsilon^+) + (1 - y_{i,j}) \cdot max(0, \epsilon^- - d(h_i, h_j)), \tag{7}$$

to train our model. $y_{i,j}$ is 1 for aligned and 0 for non-aligned pairs of nodes, and d is the Euclidean distance. In contrast to, e.g., triplet loss, the contrastive loss allows us to incorporate nodes having no matchings into the training.

3 Experiments

We evaluate our approach by interpreting the node matching problem as a binary classification task and compare the proposed GrAN model against several state-of-the-art methods including GCN [20], GAT [21], and GraphSAGE [22]. All GNN models use the same siamese architecture with shared weights. For the competitors, as well as for our model we use implementations from the pytorch geometric framework [23]. We also evaluate against a baseline method where we simply use the geo-spatial coordinates of the used road networks' vertices as 2-dimensional node representations for the evaluation. Note, that this is a fairly strong baseline as the graph vertices lie within a continuous space and two matching nodes obviously tend to have similar geo-spatial coordinates. Additionally, where it is possible, we report the results when using probabilistic relaxation [14]. Note, that the probabilistic relaxation method considers all assignments at once using hungarian algorithm [24] in combination with Otsu's [25] method, and therefore can benefit from the fact that graphs are different in size. However, the complexity of $O(c^3)$, with c being the number of matching candidates, is rather high and hence makes it impractical for larger networks. In contrast, all GNN models classify each candidate pair independently. To achieve a fair comparison, we additionally report results when conducting nearest neighbor queries on the set of nodes from the larger graph and query objects only being taken from the smaller graph. This way we still need to find a threshold to divide the retrieved candidates into nodes with and without matchings, but consider the fact that graphs differ in size. Note that this approach ignores cases where nodes match to multiple nodes in counterpart graph.

We train each of the models by using two different road networks (one is from OSM[1] and the other from a commercial map provider), that both cover freeways of the Northrhine-Westphalia region in Germany. In total, the networks consist of 15,442 vertices and 20,166 edges with 2956 of the vertices having a matching node in the other network, i.e., they form matching pairs. In general, nodes can have zero (1:0), one (1:1) or multiple (1:n) matching nodes in the other graph. Every node represents an intersection and each edge corresponds to a road segment represented by the sequence

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

of coordinates. To train the models, the matching pairs are split into training, validation and test set (80-10-10). For the purpose of generating negative samples, we project each of the positive sample vertices into the other road network and perform spatial range queries with a radius of approximately 1km^2. The resulting set of vertices form the negative samples of the corresponding node (except for the actual matching vertex). Additionally, we perform the same range queries for the nodes labeled as 1:0 matchings and add the resulting pairs to the negative samples. Finally we get a set of 65,432 positive and negative candidate pairs, that form the final training, validation and test sets as reported in Table 3 in supplementary material. Furthermore, we evaluate how well the models generalize to yet unseen road networks by measuring their matching performance on a pair of road network sections that are taken from the urban area of the city of Munich, Germany. Beside freeways, these graphs also contain types of roads that are not present in the training data, e.g., residential street. The dataset is referred to as MUC. For all models we trained various different architectures whose details and hyperparameter settings can be found in the supplementary material. In the following, we report the results for the architectures that showed the best performance.

Table 1: Resulting F1 Scores for the NRW and MUC datasets.

Method	**NRW**	**MUC**
GrAN	.676	.642
GCN	.091	.009
GAT	.091	.009
GraphSAGE	.091	.009
Spatial Coordinates	.681	.626
Probabilistic Relaxation	.752	-

Table 2: Resulting F1 Scores for the NRW and MUC datasets when using NN queries.

Method	**NRW**	**MUC**
GrAN	.847	.704
GCN	.048	.001
GAT	.071	.005
GraphSAGE	.064	.001
Spatial Coordinates	.827	.661
Probabilistic Relaxation	.752	-

Table 1 shows the F1 scores comparing the classification results of our approach against the results produced by the other GNN based models and when using only spatial coordinates (SC), or the probabilistic relaxation method. While our approach outperforms GCN, GAT and GraphSage, and also the SC baseline on the MUC dataset, we achieve similar matching results as the baseline on the NRW data. When comparing against probabilistic relaxation, the latter achieves better results on the NRW dataset. However, considering the nearest neighbors evaluation (cf. Table 2), the probabilistic relaxation approach is outperformed by both our approach and SC, although all methods except for probabilistic relaxation ignore 1:n matchings. In summary, our method still achieves best results on both datasets.

4 Conclusion

In this work we studied the problem of graph alignment. We presented our ongoing work on a new Graph Neural Network based model, that aggregates information from neighbors based on their alignment scores. We evaluated the proposed approach on Map Fusion tasks and compared it with state-of-the-art models. Our promising experimental results show that the proposed approach outperforms other GNN models by large margins. In future work we plan to adapt our model for other tasks like the alignment of entities in heterogeneous graphs, and also for calculating graph similarities. We further see the potential of improving the runtime of our algorithm by combining it with graph coarsening methods for instance.

Acknowledgements

This work has partially been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A, and has been developed in cooperation with Harman International. The authors of this work take full responsibilities for its content.

[2]Note that some matching nodes representing the same acceleration lanes or freeway exits are indeed located that far from each other.

References

[1] Juan J Ruiz, F Javier Ariza, Manuel A Urena, and Elidia B Blázquez. Digital map conflation: a review of the process and a proposal for classification. *International Journal of Geographical Information Science*, 25(9):1439–1466, 2011.

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

[3] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. Semi-supervised entity alignment via knowledge graph embedding with awareness of degree difference. In *The World Wide Web Conference*, pages 3130–3136. ACM, 2019.

[4] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. Multi-view knowledge graph embedding for entity alignment. *arXiv preprint arXiv:1906.02390*, 2019.

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

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

[7] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. Neighborhood-aware attentional representation for multilingual knowledge graphs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1943–1949. International Joint Conferences on Artificial Intelligence Organization, 2019.

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

[9] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. Cross-lingual knowledge graph alignment via graph matching neural network. *arXiv preprint arXiv:1905.11605*, 2019.

[10] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Juanzi Li, and Tat-Seng Chua. Multi-channel graph neural network for entity alignment. *arXiv preprint arXiv:1908.09898*, 2019.

[11] Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *IJCAI*, pages 1774–1780, 2016.

[12] Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. *arXiv preprint arXiv:1904.12787*, 2019.

[13] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 384–392. ACM, 2019.

[14] Bisheng Yang, Yunfei Zhang, and Xuechen Luan. A probabilistic relaxation approach for matching road networks. *International Journal of Geographical Information Science*, 27(2):319–338, 2013.

[15] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[16] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[17] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

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

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

[22] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[23] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[24] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[25] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[26] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

5 Supplementary material

5.1 Dataset statistics

Table 3: Training and test dataset statistics.

	NRW (train)	NRW (test)	MUC (test)
Nodes in G_1	2,430	308	2,996
Nodes in G_2	9,924	1,206	3,153
Candidates	52,138	6,148	378,948
1:1 & 1:n matchings	2,359	294	1,777
1:0 matchings	99	3	1,257

5.2 Architectures & Hyperparameter Settings

For all Graph Neural Network models we tried different architectures and hyperparameters:

Weight Sharing in Siamese Architecture We also tried to learn different encoders for each map without weight sharing. This could help if two maps have different biases. However, it only worsened the results.

Unsupervised Pretraining We used an adapted version of Deep Graph Infomax [26] for pretraining. To corrupt a graph we shifted nodes randomly in coordinate space in each iteration. Model pretraining led to the fastest convergence but did not affect a final result significantly.

Loss We experimented with different positive and negative margins. We obtained best results with a positive margin of 0 and a negative margin of 10. To account for the class imbalance, we decreased the weights for the negative examples inversely proportional to the relative frequency.

Model Depth We tried 1 to 8 message passing layers, followed by 0 to 2 fully connected layers. For GAT we additionally tried 1 to 7 attention heads in each layer. The number of attention heads was constant for all layers.

Width We tested various layer sizes, with both constant and variable sizes of layers in the same model.

Edge Attributes For the fair comparison we adapted the message function of all GNN models to be able to consider edge attributes and we evaluated them with and without edge attributes. We also trained models which consider edge attributes only in the first layer. Using the Douglas–Peucker algorithm we reduced edge attribute sequence to different lengths. We tried the sum and concatenation operations for combining edge and node embeddings in the function M.

Early Stopping We trained all models with early stopping using a patience of 1000.

Message Aggregation To aggregate messages, we tried mean/max pooling and concatenation.

Normalization We normalized the input coordinates to [-1,1] range. Therefore, the data are first zero-centered for each dimension. Next, each dimension is divided by the max distance from the center in all dimensions. This normalization stabilizes training and inference while preserving distance ratios.

Optimizer Settings All tested models were trained with the Adam SGD optimizer with learning rate $lr \in [1\text{e-}4, 1\text{e-}2)$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the weight decay being chosen from the interval $[0, 1\text{e-}4)$

Chapter 7

Active Learning for Entity Alignment

The chapter includes following publication (omitted due to copyright reasons):

Max Berrendorf*, Evgeniy Faerman*, and Volker Tresp. "Active Learning for Entity Alignment". In: *Advances in Information Retrieval*. *equal contribution. Springer International Publishing, 2021, pp. 48–62. DOI: 10.1007/978-3-030-72113-8_4

Chapter 8

On the Ambiguity of Rank-Based Evaluation of Entity Alignment or Link Prediction Methods

The chapter includes following publication (omitted due to copyright reasons):

Max Berrendorf, Evgeniy Faerman, Laurent Vermue, and Volker Tresp. "Interpretable and Fair Comparison of Link Prediction or Entity Alignment Methods with Adjusted Mean Rank". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020), pp. 363–366

Chapter 9

Knowledge Graph Entity Alignment with Graph Convolutional Networks: Lessons Learned

The chapter includes following publication (omitted due to copyright reasons):

Max Berrendorf, Evgeniy Faerman, Valentyn Melnychuk, Volker Tresp, and Thomas Seidl. "Knowledge graph entity alignment with graph convolutional networks: Lessons learned". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 3–11. DOI: 10.1007/978-3-030-45442-5_1

Chapter 10

A Critical Assessment of State-of-the-Art in Entity Alignment

The chapter includes following publication (omitted due to copyright reasons):

Max Berrendorf, Evgeniy Faerman, Valentyn Melnychuk, Volker Tresp, and Thomas Seidl. "Knowledge graph entity alignment with graph convolutional networks: Lessons learned". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 3–11. DOI: 10.1007/978-3-030-45442-5_1

Chapter 11

TACAM: Topic And Context Aware Argument Mining

The chapter includes following publication (omitted due to copyright reasons):

Michael Fromm, Evgeniy Faerman, and Thomas Seidl. "TACAM: Topic And Context Aware Argument Mining". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2019, pp. 99–106. DOI: 10.1145/3350546.3352506

Chapter 12

Argument Mining Driven Analysis of Peer-Reviews

The chapter includes following publication (omitted due to copyright reasons):

Fromm Michael, Faerman Evgeniy, Berrendorf Max, Bhargava Siddharth, Qi Ruoxia, Zhang Yao, Dennert Lukas, Selle Sophia, Mao Yang, and Seidl Thomas. "Argument Mining Driven Analysis of Peer-Reviews". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021

Chapter 13

Diversity Aware Relevance Learning for Argument Search

The chapter includes following publication (omitted due to copyright reasons):

Michael Fromm, Max Berrendorf, Sandra Obermeier, Thomas Seidl, and Evgeniy Faerman. "Diversity Aware Relevance Learning for Argument Search". In: *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 264–271. DOI: 10.1007/978-3-030-72240-1_24

Chapter 14

Learning Self-Expression Metrics for Scalable and Inductive Subspace Clustering

The chapter includes following publication (omitted due to copyright reasons):

Julian Busch, Evgeniy Faerman, Matthias Schubert, and Thomas Seidl. "Learning Self-Expression Metrics for Scalable and Inductive Subspace Clustering". In: *NeurIPS 2020 Workshop on Self-Supervised Learning: Theory and Practice* (2020). arXiv: 2009.12875 [cs.LG]

Bibliography

[AA19] Waseem Ahmad and Rashid Ali. "Social account matching in online social media using cross-linked posts". In: *Procedia Computer Science* 152 (2019), pp. 222–229.

[ACL06] Reid Andersen, Fan Chung, and Kevin Lang. "Local graph partitioning using pagerank vectors". In: *Proc. of IEEE FOCS*. IEEE. 2006, pp. 475–486.

[AEH+17] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. "Watch your step: Learning graph embeddings through attention". In: *arXiv preprint arXiv:1710.09599* (2017).

[Ahm+13] Amr Ahmed, Nino Shervashidze, Shravan Narayanamurthy, Vanja Josifovski, and Alexander J Smola. "Distributed large-scale natural graph factorization". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 37–48.

[AT16] James Atwood and Don Towsley. "Diffusion-convolutional neural networks". In: *Advances in neural information processing systems*. 2016, pp. 1993–2001.

[Bat+18] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[BCB14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473* (2014).

[BCM11] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. "Node classification in social networks". In: *Social network data analytics*. Springer, 2011, pp. 115–148.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[Ber+20a] Max Berrendorf, Evgeniy Faerman, Valentyn Melnychuk, Volker Tresp, and Thomas Seidl. "Knowledge graph entity alignment with graph convolutional networks: Lessons learned". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 3–11. DOI: 10.1007/978-3-030-45442-5_1.

[Ber+20b] Max Berrendorf, Evgeniy Faerman, Laurent Vermue, and Volker Tresp. "Interpretable and Fair Comparison of Link Prediction or Entity Alignment Methods with Adjusted Mean Rank". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020), pp. 363–366.

[BFT21] Max Berrendorf*, Evgeniy Faerman*, and Volker Tresp. "Active Learning for Entity Alignment". In: *Advances in Information Retrieval*. *equal contribution. Springer International Publishing, 2021, pp. 48–62. DOI: 10.1007/978-3-030-72113-8_4.

[BG07] Indrajit Bhattacharya and Lise Getoor. "Collective entity resolution in relational data". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1.1 (2007), 5–es.

[BMB10] Jamal Bentahar, Bernard Moulin, and Micheline Bélanger. "A taxonomy of argumentation models used for knowledge representation". In: *Artificial Intelligence Review* 33.3 (2010), pp. 211–259.

[Bor+13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.

[Bor+19] Felix Borutta, Julian Busch*, Evgeniy Faerman*, Adina Klink, and Matthias Schubert. "Structural Graph Representations based on Multiscale Local Network Topologies". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. *equal contribution. IEEE. 2019, pp. 91–98. DOI: 10.1145/3350546.3352505.

[BP98] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine". In: (1998).

[Bro+17] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. "Geometric deep learning: going beyond euclidean data". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[Bro+20] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[Bro+94] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems*. 1994, pp. 737–744.

[Bru+13] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. "Spectral Networks and Locally Connected Networks on Graphs". In: *CoRR* abs/1312.6203 (2013). URL: http://arxiv.org/abs/1312.6203.

[BTM18] Florian Bernard, Christian Theobalt, and Michael Moeller. "DS*: Tighter lifting-free convex relaxations for quadratic matching problems". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4310–4319.

[Bus+20] Julian Busch, Evgeniy Faerman, Matthias Schubert, and Thomas Seidl. "Learning Self-Expression Metrics for Scalable and Inductive Subspace Clustering". In: *NeurIPS 2020 Workshop on Self-Supervised Learning: Theory and Practice* (2020). arXiv: 2009.12875 [cs.LG].

[BVh19] Wim Boes and Hugo Van hamme. "Audiovisual Transformer Architectures for Large-Scale Classification and Synchronization of Weakly Labeled Audio Events". In: *Proceedings of the 27th ACM International Conference on Multimedia*. 2019, pp. 1961–1969.

[BWF21] Max Berrendorf, Ludwig Wacker, and Evgeniy Faerman. "A Critical Assessment of State-of-the-Art in Entity Alignment". In: *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 18–32. DOI: 10.1007/978-3-030-72240-1_2.

[Cae+09] Tibério S Caetano, Julian J McAuley, Li Cheng, Quoc V Le, and Alex J Smola. "Learning graph matching". In: *IEEE transactions on pattern analysis and machine intelligence* 31.6 (2009), pp. 1048–1058.

[Can13] Yaiza Canzani. "Analysis on manifolds via the Laplacian". In: *Lecture Notes available at: http://www. math. harvard. edu/canzani/docs/Laplacian. pdf* (2013).

[Can+18] Cătălina Cangea, Petar Veličković, Nikola Jovanović, Thomas Kipf, and Pietro Liò. "Towards sparse hierarchical graph classifiers". In: *arXiv preprint arXiv:1811.01287* (2018).

[Cao+19] Yixin Cao, Zhiyuan Liu, Chengjiang Li, Zhiyuan Liu, Juanzi Li, and Tat-Seng Chua. "Multi-Channel Graph Neural Network for Entity Alignment". In: *ACL (1)*. ACL, 2019, pp. 1452–1461.

[CAP13] Minsu Cho, Karteek Alahari, and Jean Ponce. "Learning graphs to match". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 25–32.

[CDI98] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. "Enhanced hypertext categorization using hyperlinks". In: *Acm Sigmod Record* 27.2 (1998), pp. 307–318.

[Che+17] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. "Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment". In: *IJCAI*. ijcai.org, 2017, pp. 1511–1517.

[Che+18] Muhao Chen, Yingtao Tian, Kai-Wei Chang, Steven Skiena, and Carlo Zaniolo. "Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment". In: *arXiv preprint arXiv:1806.06478* (2018).

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[CK10] Michael Chertok and Yosi Keller. "Efficient high order matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), pp. 2205–2215.

[CLL10] Minsu Cho, Jungmin Lee, and Kyoung Mu Lee. "Reweighted random walks for graph matching". In: *European conference on Computer vision*. Springer. 2010, pp. 492–505.

[CLX15] Shaosheng Cao, Wei Lu, and Qiongkai Xu. "Grarep: Learning graph representations with global structural information". In: *Proc. of CIKM*. ACM. 2015, pp. 891–900.

[CLX16] Shaosheng Cao, Wei Lu, and Qiongkai Xu. "Deep neural networks for learning graph representations". In: *Thirtieth AAAI conference on artificial intelligence*. 2016.

[Con+17] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. "Word translation without parallel data". In: *arXiv preprint arXiv:1710.04087* (2017).

[Cor+20] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. "Principal neighbourhood aggregation for graph nets". In: *arXiv preprint arXiv:2004.05718* (2020).

[CT15] Lucas Carstens and Francesca Toni. "Towards relation based argumentation mining". In: *Proceedings of the 2nd Workshop on Argumentation Mining*. 2015, pp. 29–34.

[CT17] Oana Cocarascu and Francesca Toni. "Identifying attack and support argumentative relations using deep learning". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1374–1379. DOI: 10.18653/v1/D17-1144. URL: https://www.aclweb.org/anthology/D17-1144.

[CV18] Elena Cabrio and Serena Villata. "Five Years of Argument Mining: a Data-driven Analysis." In: *IJCAI*. Vol. 18. 2018, pp. 5427–5433.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. 2016, pp. 3844–3852.

[DBY19] Nima Dehmamy, Albert-László Barabási, and Rose Yu. "Understanding the representation power of graph neural networks in learning graph topology". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15413–15423.

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[DNS20] Lorik Dumani, Patrick J Neumann, and Ralf Schenkel. "A Framework for Argument Retrieval". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 431–445.

[Don+14] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. "Knowledge vault: A web-scale approach to probabilistic knowledge fusion". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 601–610.

[Don+18] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. "Learning structural node embeddings via diffusion wavelets". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 1320–1329.

[DS19] Lorik Dumani and Ralf Schenkel. "A systematic comparison of methods for finding good premises for claims". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 957–960.

[Duv+15] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. "Convolutional networks on graphs for learning molecular fingerprints". In: *Advances in neural information processing systems*. 2015, pp. 2224–2232.

[Dwi+20] Vijay Prakash Dwivedi, Chaitanya K Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. "Benchmarking graph neural networks". In: *arXiv preprint arXiv:2003.00982* (2020).

[EBT19] Pantelis Elinas, Edwin V Bonilla, and Louis Tiao. "Variational Inference for Graph Convolutional Networks in the Absence of Graph Data and Adversarial Settings". In: *arXiv preprint arXiv:1906.01852* (2019).

[ED+20] Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, et al. "Corpus Wide Argument Mining-A Working Solution." In: *AAAI*. 2020, pp. 7683–7691.

[EDG17] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. "Neural end-to-end learning for computational argumentation mining". In: *arXiv preprint arXiv:1704.06104* (2017).

[Fae+18] Evgeniy Faerman, Felix Borutta, Kimon Fountoulakis, and Michael W Mahoney. "Lasagne: Locality and structure aware graph node embedding". In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. **Best Student Paper Award**. IEEE. 2018, pp. 246–253. DOI: 10.1109/WI.2018.00-83.

[Fae+19a] Evgeniy Faerman, Manuell Rogalla, Niklas Strauß, Adrian Krüger, Benedict Blümel, Max Berrendorf, Michael Fromm, and Matthias Schubert. "Spatial Interpolation with Message Passing Framework". In: *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2019, pp. 135–141. DOI: 10.1109/ICDMW.2019.00030.

[Fae+19b] Evgeniy Faerman, Otto Voggenreiter, Felix Borutta, Tobias Emrich, Max Berrendorf, and Matthias Schubert. "Graph Alignment Networks with Node Matching Scores". In: *Graph Representation Learning NeurIPS 2019 Workshop*. 2019.

[Fae+20] Evgeniy Faerman, Felix Borutta, Julian Busch, and Matthias Schubert. "Ada-LLD: Adaptive Node Similarity for Node Classification Using Multi-Scale Local Label Distributions". In: *The 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'20)* (2020). **Best Student Paper Award**, pp. 25–32.

[Fen+20] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. "Language-agnostic BERT Sentence Embedding". In: *arXiv preprint arXiv:2007.01852* (2020).

[Fey19] Matthias Fey. "Just jump: Dynamic neighborhood aggregation in graph neural networks". In: *arXiv preprint arXiv:1904.04849* (2019).

[Fey+20] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. "Deep graph matching consensus". In: *arXiv preprint arXiv:2001.09621* (2020).

[FFS19] Michael Fromm, Evgeniy Faerman, and Thomas Seidl. "TACAM: Topic And Context Aware Argument Mining". In: *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2019, pp. 99–106. DOI: 10.1145/3350546.3352506.

[FL19] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[Fro+21] Michael Fromm, Max Berrendorf, Sandra Obermeier, Thomas Seidl, and Evgeniy Faerman. "Diversity Aware Relevance Learning for Argument Search". In: *Advances in Information Retrieval*. Cham: Springer International Publishing, 2021, pp. 264–271. DOI: 10.1007/978-3-030-72240-1_24.

[Gal+20] Andrea Galassi, Kristian Kersting, Marco Lippi, Xiaoting Shao, and Paolo Torroni. "Neural-symbolic argumentation mining: An argument in favor of deep learning and reasoning". In: *Frontiers in Big Data* 2 (2020), p. 52.

[Gat+15] Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, and Christos Faloutsos. "Linearized and single-pass belief propagation". In: *VLDB Endowment* 8.5 (2015), pp. 581–592.

[GERD13] Sean Gilpin, Tina Eliassi-Rad, and Ian Davidson. "Guided learning for role discovery (GLRD) framework, algorithms, and applications". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2013, pp. 113–121.

[GG84] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[GH12] Michael U Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics". In: *Journal of Machine Learning Research* 13.Feb (2012), pp. 307–361.

[Gil+17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. "Neural message passing for quantum chemistry". In: *arXiv preprint arXiv:1704.01212* (2017).

[GL16] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: *Proc. of ACM SIGKDD*. 2016, pp. 855–864.

[Gle15] David F Gleich. "PageRank beyond the Web". In: *Siam Review* 57.3 (2015), pp. 321–363.

[Gle+19] Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkowich, Ranit Aharonov, and Noam Slonim. "Are You Convinced? Choosing the More Convincing Evidence with a Siamese Network". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 967–976. DOI: 10.18653/v1/P19-1093. URL: https://www.aclweb.org/anthology/P19-1093.

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[GR96] Steven Gold and Anand Rangarajan. "A graduated assignment algorithm for graph matching". In: *IEEE Transactions on pattern analysis and machine intelligence* 18.4 (1996), pp. 377–388.

[Gre+20] Shai Gretz, Roni Friedman, Edo Cohen-Karlik, Assaf Toledo, Dan Lahav, Ranit Aharonov, and Noam Slonim. "A large-scale dataset for argument quality ranking: Construction and analysis". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 7805–7813.

[Guo+18] Michelle Guo, Edward Chou, De-An Huang, Shuran Song, Serena Yeung, and Li Fei-Fei. "Neural graph matching networks for fewshot 3d action recognition". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 653–669.

[Hab+18] Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. "Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation". In: *arXiv preprint arXiv:1802.06613* (2018).

[Hao+16] Yanchao Hao, Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. "A joint embedding method for entity alignment of knowledge bases". In: *China Conference on Knowledge Graph and Semantic Computing*. Springer. 2016, pp. 3–14.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[Hei+18] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. "Regal: Representation learning-based graph alignment". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 117–126.

[HEKG14] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. "Argumentation Mining on the Web from Information Seeking Perspective." In: *ArgNLP*. 2014.

[Hen+11] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. "It's who you know: graph mining using recursive structural features". In: *Proc. of ACM SIGKDD*. 2011, pp. 663–671.

[Hen+12] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. "Rolx: structural role extraction & mining in large graphs". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 1231–1239.

[HG16] Ivan Habernal and Iryna Gurevych. "Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2016, pp. 1589–1599.

[HG17] Ivan Habernal and Iryna Gurevych. "Argumentation mining in user-generated web discourse". In: *Computational Linguistics* 43.1 (2017), pp. 125–179.

[Him+16] Itai Himelboim, Kaye D Sweetser, Spencer F Tinkham, Kristen Cameron, Matthew Danelo, and Kate West. "Valence-based homophily on Twitter: Network analysis of emotions and political talk in the 2012 presidential election". In: *New media & society* 18.7 (2016), pp. 1382–1400.

[Hje+18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. "Learning deep representations by mutual information estimation and maximization". In: *arXiv preprint arXiv:1808.06670* (2018).

[HL81] Paul W Holland and Samuel Leinhardt. "An exponential family of probability distributions for directed graphs". In: *Journal of the american Statistical association* 76.373 (1981), pp. 33–50.

[Hu+20] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. "Open graph benchmark: Datasets for machine learning on graphs". In: *arXiv preprint arXiv:2005.00687* (2020).

[HVG11] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on graphs via spectral graph theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *NIPS*. 2017, pp. 1025–1035.

[IMG17] Umar Iqbal, Anton Milan, and Juergen Gall. "Posetrack: Joint multi-person pose estimation and tracking". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2011–2020.

[JLH11] Ruoming Jin, Victor E Lee, and Hui Hong. "Axiomatic ranking of network role similarity". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 922–930.

[JLL14] Ruoming Jin, Victor E Lee, and Longjie Li. "Scalable and axiomatic ranking of network role similarity". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8.1 (2014), pp. 1–37.

[JNG04] David Jensen, Jennifer Neville, and Brian Gallagher. "Why collective inference improves relational classification". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, pp. 593–598.

[Joa03] Thorsten Joachims. "Transductive learning via spectral graph partitioning". In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003, pp. 290–297.

[Kaz+20] Anees Kazi, Luca Cosmo, Nassir Navab, and Michael Bronstein. "Differentiable Graph Module (DGM) Graph Convolutional Networks". In: *arXiv preprint arXiv:2002.04999* (2020).

[KBG18] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. "Predict then propagate: Graph neural networks meet personalized pagerank". In: *arXiv preprint arXiv:1810.05997* (2018).

[KFL01] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Transactions on information theory* 47.2 (2001), pp. 498–519.

[KK14] Isabel M Kloumann and Jon M Kleinberg. "Community membership identification from small seed sets". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 1366–1375.

[Kle99] Jon M Kleinberg. "Authoritative sources in a hyperlinked environment". In: *Journal of the ACM (JACM)* 46.5 (1999), pp. 604–632.

[Kni08] Philip A Knight. "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (2008), pp. 261–275.

[Kou+11] Danai Koutra, Tai-You Ke, U Kang, Duen Horng Polo Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. "Unifying guilt-by-association approaches: Theorems and fast algorithms". In: *Proc. of ECML PKDD*. Springer. 2011, pp. 245–260.

[Kuh55] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[KW16a] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[KW16b] Thomas N Kipf and Max Welling. "Variational graph auto-encoders". In: *arXiv preprint arXiv:1611.07308* (2016).

[KWG19] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. "Diffusion Improves Graph Learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13333–13345.

[Lau+20] Anne Lauscher, Lily Ng, Courtney Napoles, and Joel Tetreault. *Rhetoric, Logic, and Dialectic: Advancing Theory-based Argument Quality Assessment in Natural Language Processing*. 2020. arXiv: `2006.00843 [cs.CL]`.

[Law63] Eugene L Lawler. "The quadratic assignment problem". In: *Management science* 9.4 (1963), pp. 586–599.

[LeC+99] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. "Object recognition with gradient-based learning". In: *Shape, contour and grouping in computer vision*. Springer, 1999, pp. 319–345.

[Les+09] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters". In: *Internet Mathematics* 6.1 (2009), pp. 29–123.

[Lev+14] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. "Context dependent claim detection". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2014, pp. 1489–1500.

[LF92] D Lenat and E Feigenbaum. "On the thresholds of knowledge". In: *Foundations of Artificial Intelligence, MIT Press, Cambridge, MA* (1992), pp. 185–250.

[LG03] Qing Lu and Lise Getoor. "Link-Based Classification". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. Washington, DC, USA: AAAI Press, 2003, 496–503. ISBN: 1577351894.

[LG14] Omer Levy and Yoav Goldberg. "Neural word embedding as implicit matrix factorization". In: *Advances in neural information processing systems*. 2014, pp. 2177–2185.

[LH05] Marius Leordeanu and Martial Hebert. "A spectral technique for correspondence problems using pairwise constraints". In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1482–1489.

[Li+16] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. "Gated Graph Sequence Neural Networks". In: *ICLR*. 2016.

[Li+19a] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, and Tat-Seng Chua. "Semi-supervised Entity Alignment via Joint Knowledge Embedding Model and Cross-graph Model". In: *EMNLP/IJCNLP (1)*. ACL, 2019, pp. 2723–2732.

[Li+19b] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. "Deepgcns: Can gcns go as deep as cnns?" In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 9267–9276.

[Li+20] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. "DeeperGCN: All You Need to Train Deeper GCNs". In: *arXiv preprint arXiv:2006.07739* (2020).

[Lin+19] Xixun Lin, Hong Yang, Jia Wu, Chuan Zhou, and Bin Wang. "Guiding cross-lingual entity alignment via adversarial knowledge embedding". In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2019, pp. 429–438.

[LLS16] Yankai Lin, Zhiyuan Liu, and Maosong Sun. "Knowledge representation learning with entities, attributes and relations". In: *ethnicity* 1 (2016), pp. 41–52.

[LR20] John Lawrence and Chris Reed. "Argument mining: A survey". In: *Computational Linguistics* 45.4 (2020), pp. 765–818.

[LSH12] Marius Leordeanu, Rahul Sukthankar, and Martial Hebert. "Unsupervised learning for graph matching". In: *International journal of computer vision* 96.1 (2012), pp. 28–45.

[LT16a] Marco Lippi and Paolo Torroni. "Argument Mining from Speech: Detecting Claims in Political Debates." In: *AAAI*. Vol. 16. 2016, pp. 2979–2985.

[LT16b] Marco Lippi and Paolo Torroni. "Argumentation mining: State of the art and emerging trends". In: *ACM Transactions on Internet Technology (TOIT)* 16.2 (2016), pp. 1–25.

[Luc+03] Joseph J Luczkovich, Stephen P Borgatti, Jeffrey C Johnson, and Martin G Everett. "Defining and measuring trophic role similarity in food webs using regular equivalence". In: *Journal of Theoretical Biology* 220.3 (2003), pp. 303–321.

[LW71] Francois Lorrain and Harrison C White. "Structural equivalence of individuals in social networks". In: *The Journal of mathematical sociology* 1.1 (1971), pp. 49–80.

[Mao+20] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. "MRAEA: An Efficient and Robust Entity Alignment Approach for Cross-lingual Knowledge Graph". In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. 2020, pp. 420–428.

[MD13] T Mikolov and J Dean. "Distributed representations of words and phrases and their compositionality". In: *NIPS* (2013).

[MH15] Yann LeCun Mikael Henaff Joan Bruna. "Deep Convolutional Networks on Graph-Structured Data". In: *arXiv preprint arXiv:1506.05163* (2015).

[Mic+21] Fromm Michael, Faerman Evgeniy, Berrendorf Max, Bhargava Siddharth, Qi Ruoxia, Zhang Yao, Dennert Lukas, Selle Sophia, Mao Yang, and Seidl Thomas. "Argument Mining Driven Analysis of Peer-Reviews". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021.

[Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[Mit80] Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ..., 1980.

[MNN19] Sameh K Mohamed, Aayah Nounu, and Vít Nováček. "Drug target discovery using knowledge graph embeddings". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 2019, pp. 11–18.

[Mor+19] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. "Weisfeiler and leman go neural: Higher-order graph neural networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4602–4609.

[MP03] Sofus A Macskassy and Foster Provost. *A simple relational classifier*. Tech. rep. 2003.

[MP07] Sofus A Macskassy and Foster Provost. "Classification in networked data: A toolkit and a univariate case study". In: *Journal of machine learning research* 8.May (2007), pp. 935–983.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[Mur+18] Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. "Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs". In: *arXiv preprint arXiv:1811.01900* (2018).

[NH15] Nona Naderi and Graeme Hirst. "Argumentation mining in parliamentary discourse". In: *Principles and practice of multi-agent systems*. Springer, 2015, pp. 16–25.

[Nic+15] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. "A review of relational machine learning for knowledge graphs". In: *Proceedings of the IEEE* 104.1 (2015), pp. 11–33.

[NJ00] Jennifer Neville and David Jensen. "Iterative classification in relational data". In: *Proc. AAAI-2000 workshop on learning statistical models from relational data*. 2000, pp. 13–20.

[NKG15] Huda Nassar, Kyle Kloster, and David F Gleich. "Strong localization in personalized PageRank vectors". In: *International Workshop on Algorithms and Models for the Web-Graph*. Springer. 2015, pp. 190–202.

[NL18] Huy V Nguyen and Diane J Litman. "Argument Mining for Improving the Automated Scoring of Persuasive Essays." In: *AAAI*. Vol. 18. 2018, pp. 5892–5899.

[NM19] Hoang NT and Takanori Maehara. "Revisiting graph neural networks: All we have is low-pass filters". In: *arXiv preprint arXiv:1905.09550* (2019).

[NM91] Susan Newman and Catherine Marshall. "Pushing Toulmin too far: Learning from an argument representation scheme". In: *Xerox PARC, Palo Alto, CA, USA, Technical Report SSL-92* 45 (1991).

[Now+17] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. "A note on learning algorithms for quadratic assignment with graph neural networks". In: *Proceeding of the 34th International Conference on Machine Learning (ICML)*. Vol. 1050. 2017, p. 22.

[Now+18] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. "Revised note on learning quadratic assignment with graph neural networks". In: *2018 IEEE Data Science Workshop (DSW)*. IEEE. 2018, pp. 1–5.

[Noy+19] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. "Industry-scale knowledge graphs: lessons and challenges". In: *Queue* 17.2 (2019), pp. 48–75.

[Orb+20] Matan Orbach, Yonatan Bilu, Assaf Toledo, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. "Out of the Echo Chamber: Detecting Countering Debate Speeches". In: *arXiv preprint arXiv:2005.01157* (2020).

[Pag+99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The PageRank citation ranking: bringing order to the web." In: (1999).

[Par+18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. "Image transformer". In: *arXiv preprint arXiv:1802.05751* (2018).

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proc. of ACM SIGKDD*. 2014, pp. 701–710.

[Pea82] Judea Pearl. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.

[Pee17] Leto Peel. "Graph-based semi-supervised learning for relational networks". In: *Proc. of SDM*. SIAM. 2017, pp. 435–443.

[Pei+19] Shichao Pei, Lu Yu, Robert Hoehndorf, and Xiangliang Zhang. "Semi-Supervised Entity Alignment via Knowledge Graph Embedding with Awareness of Degree Difference". In: *WWW*. ACM, 2019, pp. 3130–3136.

[PHG15] Maria Pershina, Yifan He, and Ralph Grishman. "Personalized page rank for named entity disambiguation". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2015, pp. 238–243.

[PM09] Raquel Mochales Palau and Marie-Francine Moens. "Argumentation mining: the detection, classification and structure of arguments in text". In: *Proceedings of the 12th international conference on artificial intelligence and law*. 2009, pp. 98–107.

[Pot+19] Martin Potthast, Lukas Gienapp, Florian Euchner, Nick Heilenkötter, Nico Weidmann, Henning Wachsmuth, Benno Stein, and Matthias Hagen. "Argument search: assessing argument relevance". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, pp. 1117–1120.

[PP03] Claudia Perlich and Foster Provost. "Aggregation-based feature invention and relational concept classes". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, pp. 167–176.

[Qas+19] Shah Rukh Qasim, Jan Kieseler, Yutaro Iiyama, and Maurizio Pierini. "Learning representations of irregular particle-detector geometry with distance-weighted graph networks". In: *The European Physical Journal C* 79.7 (2019), pp. 1–11.

[Qiu+18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 2018, pp. 459–467.

[QTB19] Meng Qu, Jian Tang, and Yoshua Bengio. "Weakly-supervised knowledge graph alignment with adversarial learning". In: *arXiv preprint arXiv:1907.03179* (2019).

[RA14] Ryan A Rossi and Nesreen K Ahmed. "Role discovery in networks". In: *IEEE Transactions on Knowledge and Data Engineering* 27.4 (2014), pp. 1112–1131.

[Rad+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language models are unsupervised multitask learners". In: *OpenAI Blog* 1.8 (2019), p. 9.

[RBE10] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. "Dictionaries for sparse representation modeling". In: *Proceedings of the IEEE* 98.6 (2010), pp. 1045–1057.

[Rin+15] Ruty Rinott, Lena Dankin, Carlos Alzate, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. "Show me your evidence-an automatic method for context dependent evidence detection". In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 440–450.

[RSF17] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. "struc2vec: Learning node representations from structural identity". In: *Proc. of ACM SIGKDD*. ACM. 2017, pp. 385–394.

[Sch94] Cullen Schaffer. "A conservation law for generalization performance". In: *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 259–265.

[SDS17] Ljubisa Stankovic, Milos Dakovic, and Ervin Sejdic. "Vertex-frequency analysis: a way to localize graph spectral components [lecture notes]". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 176–182.

[Sen+08] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. "Collective classification in network data". In: *AI magazine* 29.3 (2008), p. 93.

[SEW15] Reid Swanson, Brian Ecker, and Marilyn Walker. "Argument mining: Extracting arguments from online dialogue". In: *Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue*. 2015, pp. 217–226.

[SG17] Christian Stab and Iryna Gurevych. "Parsing argumentation structures in persuasive essays". In: *Computational Linguistics* 43.3 (2017), pp. 619–659.

[SG18] Edwin Simpson and Iryna Gurevych. "Finding convincing arguments using scalable bayesian preference learning". In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 357–371.

[She01] C David Sherrill. "A Brief Review of Elementary Quantum Chemistry". In: *Georgia Institute of Technology, School of Chemistry and Biochemistry* (2001).

[SHL17] Zequn Sun, Wei Hu, and Chengkai Li. "Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding". In: *ISWC (1)*. Vol. 10587. Lecture Notes in Computer Science. Springer, 2017, pp. 628–644.

[Shu+13] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[Sin12] Amit Singhal. "Introducing the knowledge graph: things, not strings". In: *Official google blog* 5 (2012).

[Sin64] Richard Sinkhorn. "A relationship between arbitrary positive matrices and doubly stochastic matrices". In: *The annals of mathematical statistics* 35.2 (1964), pp. 876–879.

[SMG18] Christian Stab, Tristan Miller, and Iryna Gurevych. "Cross-topic argument mining from heterogeneous sources using attention-based neural networks". In: *arXiv preprint arXiv:1802.05758* (2018).

[Sta+18] Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. "Argumentext: Searching for arguments in heterogeneous sources". In: *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: demonstrations*. 2018, pp. 21–25.

[STE07] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. "Node roles and community structure in networks". In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. 2007, pp. 26–35.

[STT20] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 4498–4507.

[Sun+18] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. "Bootstrapping Entity Alignment with Knowledge Graph Embedding." In: *IJCAI*. 2018, pp. 4396–4402.

[Sun+19] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. "Knowledge graph alignment network with gated multi-hop neighborhood aggregation". In: *arXiv preprint arXiv:1911.08936* (2019).

[SX19] Xiaofei Shi and Yanghua Xiao. "Modeling multi-mapping relations for precise cross-lingual entity alignment". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 813–822.

[SXB08] Rohit Singh, Jinbo Xu, and Bonnie Berger. "Global alignment of multiple protein interaction networks with application to functional orthology detection". In: *Proceedings of the National Academy of Sciences* 105.35 (2008), pp. 12763–12768.

[Tan+15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. "Line: Large-scale information network embedding". In: *Proc. of WWW*. ACM. 2015, pp. 1067–1077.

[The+18] Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. "Attention-based graph neural network for semi-supervised learning". In: *arXiv preprint arXiv:1803.03735* (2018).

[Tia+20] Junrui Tian, Zhiying Tu, Zhongjie Wang, Xiaofei Xu, and Min Liu. "User Intention Recognition and Requirement Elicitation Method for Conversational AI Services". In: *arXiv preprint arXiv:2009.01509* (2020).

[Tol+19] Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. "Automatic Argument Quality Assessment–New Datasets and Methods". In: *arXiv preprint arXiv:1909.01007* (2019).

[Tou58] Stephen E Toulmin. *The uses of argument*. Cambridge university press, 1958.

[TQZ19] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. "Entity Alignment between Knowledge Graphs Using Attribute Embeddings". In: *AAAI*. AAAI Press, 2019, pp. 297–304.

[Tra+20] Dietrich Trautmann, Michael Fromm, Volker Tresp, Thomas Seidl, and Hinrich Schütze. "Relational and Fine-Grained Argument Mining". In: *Datenbank-Spektrum* (2020), pp. 1–7.

[TSB09] Simone Teufel, Advaith Siddharthan, and Colin Batchelor. "Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics". In: *Proceedings of the 2009 conference on empirical methods in natural language processing*. 2009, pp. 1493–1502.

[Tsi+18] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. "Verse: Versatile graph embeddings from similarity measures". In: *Proceedings of the 2018 World Wide Web Conference*. 2018, pp. 539–548.

[Tu+18] Ke Tu, Peng Cui, Xiao Wang, Philip S Yu, and Wenwu Zhu. "Deep recursive network embedding with regular equivalence". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2357–2366.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[VEGK19] Frans H Van Eemeren, Rob Grootendorst, and Tjark Kruiger. *Handbook of argumentation theory: A critical survey of classical backgrounds and modern studies*. Vol. 7. Walter de Gruyter GmbH & Co KG, 2019.

[Vel+17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[Vel+19] Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. "Neural execution of graph algorithms". In: *arXiv preprint arXiv:1910.10593* (2019).

[Wac+17] Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. "Building an argument search engine for the web". In: *Proceedings of the 4th Workshop on Argument Mining*. 2017, pp. 49–59.

[Wan+18] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. "Cross-lingual Knowledge Graph Alignment via Graph Convolutional Networks". In: *EMNLP*. ACL, 2018, pp. 349–357.

[Wan+19] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. "Dynamic graph cnn for learning on point clouds". In: *Acm Transactions On Graphics (tog)* 38.5 (2019), pp. 1–12.

[Wan+20] Fudong Wang, Nan Xue, Jin-Gang Yu, and Gui-Song Xia. "Zero-Assignment Constraint for Graph Matching with Outliers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3033–3042.

[WCZ16] Daixin Wang, Peng Cui, and Wenwu Zhu. "Structural deep network embedding". In: *Proc. of ACM SIGKDD*. ACM. 2016, pp. 1225–1234.

[WM97] David H Wolpert and William G Macready. "No free lunch theorems for optimization". In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

[Wol96] David H Wolpert. "The lack of a priori distinctions between learning algorithms". In: *Neural computation* 8.7 (1996), pp. 1341–1390.

[WSS18] Henning Wachsmuth, Shahbaz Syed, and Benno Stein. "Retrieval of the best counterargument without prior topic knowledge". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 241–251.

[WTT13] Bo Wang, Zhuowen Tu, and John K Tsotsos. "Dynamic label propagation for semi-supervised multi-class multi-label classification". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 425–432.

[Wu+19a] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. "Simplifying graph convolutional networks". In: *arXiv preprint arXiv:1902.07153* (2019).

[Wu+19b] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. "Relation-aware entity alignment for heterogeneous knowledge graphs". In: *arXiv preprint arXiv:1908.08210* (2019).

[Wu+19c] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, and Dongyan Zhao. "Jointly learning entity and relation representations for entity alignment". In: *arXiv preprint arXiv:1909.09317* (2019).

[WW20] Henning Wachsmuth and Till Werner. *Intrinsic Quality Assessment of Arguments*. 2020. arXiv: `2010.12473 [cs.CL]`.

[WYY19] Runzhong Wang, Junchi Yan, and Xiaokang Yang. "Learning combinatorial embedding networks for deep graph matching". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3056–3065.

[Xu+18a] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[Xu+18b] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. "Representation learning on graphs with jumping knowledge networks". In: *arXiv preprint arXiv:1806.03536* (2018).

[Xu+19] Kun Xu, Liwei Wang, Mo Yu, Yansong Feng, Yan Song, Zhiguo Wang, and Dong Yu. "Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network". In: *ACL (1)*. ACL, 2019, pp. 3156–3161.

[YFW03] Jonathan S Yedidia, William T Freeman, and Yair Weiss. "Understanding belief propagation and its generalizations". In: *Exploring artificial intelligence in the new millennium* 8 (2003), pp. 236–239.

[Yu+18] Tianshu Yu, Junchi Yan, Yilin Wang, Wei Liu, et al. "Generalizing graph matching beyond quadratic assignment model". In: *Advances in neural information processing systems*. 2018, pp. 853–863.

[Yu+19] Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. "Learning deep graph matching with channel-independent embedding and Hungarian attention". In: *International Conference on Learning Representations*. 2019.

[Zah+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. "Deep sets". In: *Advances in neural information processing systems*. 2017, pp. 3391–3401.

[ZG02] Xiaojin Zhu and Zoubin Ghahramani. "Learning from labeled and unlabeled data with label propagation". In: (2002).

[ZGL03] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions". In: *Proc. of ICML*. 2003, pp. 912–919.

[Zha+17] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J Smola, and Le Song. "Variational reasoning for question answering with knowledge graph". In: *arXiv preprint arXiv:1709.04071* (2017).

[Zha+19] Qingheng Zhang, Zequn Sun, Wei Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. "Multi-view Knowledge Graph Embedding for Entity Alignment". In: *IJCAI*. ijcai.org, 2019, pp. 5429–5435.

[Zho+04] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. "Learning with local and global consistency". In: *Advances in neural information processing systems*. 2004, pp. 321–328.

[Zhu+17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[Zhu+19] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. "Neighborhood-Aware Attentional Representation for Multilingual Knowledge Graphs". In: *IJCAI*. ijcai.org, 2019, pp. 1943–1949.

[ZL17] Marinka Zitnik and Jure Leskovec. "Predicting multicellular function through multi-layer tissue networks". In: *Bioinformatics* 33.14 (2017), pp. i190–i198.

[ZL19] Zhen Zhang and Wee Sun Lee. "Deep graphical feature learning for the feature matching problem". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 5087–5096.

[ZS08] Ron Zass and Amnon Shashua. "Probabilistic graph and hypergraph matching". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2008, pp. 1–8.

[ZS18] Andrei Zanfir and Cristian Sminchisescu. "Deep learning of graph matching". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2684–2693.

[ZT12] Feng Zhou and Fernando De la Torre. "Factorized graph matching". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 127–134.