Dipl.-Ing. Lukas Eberhard, BSc

Utility of Social and Narrative Features in Recommender Systems

DOCTORAL THESIS

to achieve the university degree of
Doktor der technischen Wissenschaften

submitted to
Graz University of Technology

Supervisor
Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helic
Institute of Interactive Systems and Data Science
Graz University of Technology

Graz, September 2020

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

Date, Signature

Abstract

Recommender systems have become omni-present tools that support users in everyday life tasks, such as finding products in Web stores or online movie streaming portals. For example, in online marketplaces, traditional recommender systems typically generate recommendations based on the purchase histories and profiles of users. However, we still lack insights about the usefulness of social impact, such as likes, comments, or group joins, for the task of recommending sellers to buyers in online marketplaces. There exist online discussion boards where users ask others for specific suggestions by means of descriptive narratives that contain their current recommendation needs. For example, in the domain of movies, users explain what kind of movies they are looking for by providing a free text narrative (e.g., *"Movies with minimal story, but incredible atmosphere, such as No Country for Old Men"*). Since traditional recommender systems struggle to provide relevant suggestions for such scenarios, this thesis investigates the usefulness of features extracted from narratives with the objective to automatically generate accurate suggestions. The first part of this thesis deals with the problem of aiming and understanding the usefulness of social features for recommendations on e-commerce websites, whereas the second part concerns the impact of narrative features for recommender systems. The results in this thesis demonstrate that social information from an online social or a location-based network can achieve performance at an acceptable scale, which is convenient in specific settings, whereas the highest predictability is given with features obtained from the marketplace itself. Further, the empirical analysis of a crowdsourced dataset that contains free text narratives representing movie suggestion requests from reddit demonstrates the difficulty of the narrative-driven recommendation problem. The results indicate that users on reddit mainly focus their narrative requests on positive aspects, such as movies they liked or keywords that describe the movies they would like to see. The results of the recommender experiment corroborate this finding that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power. Overall, the results contained in this thesis serve as a stepping stone for researchers and practitioners to develop new and

improve existing recommender approaches to better tackle recommendation tasks utilizing social information as well as narrative aspects.

Kurzfassung

Empfehlungssysteme sind mittlerweile Teil unseres Alltags und unterstützen Benutzer in unterschiedlichsten Aufgaben, wie zum Beispiel dem Finden von Produkten in Onlineshops oder Filmen auf Streaming-Portalen. Beispielsweise im Bereich von online Marktplätzen generieren traditionelle Empfehlungssysteme Vorschläge basierend auf bereits getätigten Käufen und Benutzerprofilen. Allerdings ist die Nützlichkeit von sozialen Aspekten, wie zum Beispiel Likes, Kommentare oder Gruppenbeitritte, für Vorhersagen von Kaufbeziehungen in online Marktplätzen noch weitgehend unerforscht. Online existieren einige Diskussionsplattformen, über die Benutzer andere nach Vorschlägen fragen, indem sie Erzählungen verfassen, die ihre aktuellen Empfehlungsbedürfnisse beschreiben. Beispielsweise im Bereich von Filmen erklären Benutzer anhand von Freitext, welche Art von Filmen sie suchen (z.B.: *„Filme mit minimaler Handlung aber unvorstellbarer Atmosphäre, wie beispielsweise No Country for Old Men“*). Nachdem traditionelle Empfehlungssysteme Schwierigkeiten haben, relevante Vorschläge für solche Szenarien zu generieren, untersucht diese Arbeit die Nützlichkeit von Merkmalen, die von Erzählungen extrahiert werden, mit dem Ziel, automatisiert akkurate Vorschläge zu generieren. Der erste Teil dieser Arbeit beschäftigt sich mit der Nützlichkeit von sozialen Aspekten für Empfehlungen im E-Commerce-Bereich, wobei der zweite Teil den Einfluss von Merkmalen aus Erzählungen für Empfehlungssysteme behandelt. Die Ergebnisse in dieser Arbeit zeigen, dass soziale Aspekte von sozialen oder standortbezogenen Netzwerken akzeptable Leistungen erzielen können, was in speziellen Situationen durchaus brauchbar sein kann. Die akkurateste Berechenbarkeit wird mittels Merkmalen erzielt, die vom Marktplatz-Netzwerk selbst extrahiert wurden. Weiters zeigt die empirische Analyse eines Crowdsourcing-Referenzdatensatzes, der Anfragen für Filmempfehlungen von Benutzern auf Reddit in Form von Freitext-Erzählungen beinhaltet, die Schwierigkeit des Problems von auf Beschreibungen basierenden Empfehlungen auf. Die Ergebnisse zeigen, dass Reddit-Benutzer die Beschreibungen ihrer Anfragen hauptsächlich auf positive Aspekte fokussieren, wie beispielsweise Filme, die sie mochten, oder Schlüsselwörter, die die Filme beschreiben, die sie gerne sehen

möchten. Die Ergebnisse des Empfehlungsexperiments untermauern die Erkenntnis, dass positive Filme und Schlüsselwörter die stärkste und negative Filmmerkmale die geringste Wirkung für akkurate Empfehlungen haben. Zusammenfassend sind die Ergebnisse dieser Arbeit für Forscher und Praktiker von großem Interesse, um neue Empfehlungsansätze zu entwickeln bzw. bestehende zu verbessern, die soziale sowie erzählende Merkmale besser in den Empfehlungsprozess integrieren können.

Acknowledgments

First of all, I would like to thank my advisor Denis Helic for his excellent academic guidance and support throughout my Ph.D. studies. Denis, I am grateful for all inspiring and motivating discussions we had. Whenever I stuck and did not know how to proceed with my research, you provided valuable advice, either in our beneficial weekly Ph.D. meetings or in informal quick conversations.

Further, I would like to thank Simon Walk for all his great support during our collaborations. His initial idea to implement a recommender bot was the foundation for our huge recommender project that covers the major part of this thesis and still offers many opportunities for future work. I often remember the varied time we had together when we shared an office.

I would also like to thank Martin Atzmüller and Christoph Trattner for the great collaboration and for teaching me so much throughout my first years as Ph.D. student.

Additionally, all my other research group colleagues and co-authors deserve my gratitude. Particularly, I want to thank Florian Geigl, Niklas Hopfgartner, Aleksandar Karakas, Patrick Kasper, Philipp Koncar, Daniel Lamprecht, Lisa Posch, Thorsten Ruprechter, Tiago Santos, and Matthias Wölbitsch for their thoughts and efforts and for the great time we had together.

Next, I want to thank all colleagues at the institute, especially all members of the "Tribeka-Runde" for the amusing daily coffee breaks. Going to work every day would not have been as fun as it was without all of you!

Finally, my deepest gratitude goes to my family and friends, who made all of this possible throughout the years as Ph.D. student. Especially, I want to thank my girlfriend Melitta who supported me the most. Without your unconditional love, these would only be blank pages!

Contents

1 Introduction **1**

1.1 Motivation . . . 1

1.2 Problem Statements, Objectives, and General Approaches 2

1.3 Research Questions . . . 6

1.4 Main Publications . . . 10

1.5 Contributions and Implications . . . 10

1.6 Structure of this Thesis . . . 11

2 Related Work **15**

2.1 Link Prediction . . . 15

2.1.1 Link Prediction Methods . . . 16

2.1.2 Link Prediction Features . . . 18

2.1.3 Extension to Previous Research . . . 20

2.2 Recommender Networks . . . 21

2.2.1 Social Information in Recommender Networks . . . 23

2.2.2 Context-Aware Recommender Systems . . . 24

2.2.3 Narrative-Driven Recommender Systems . . . 24

2.2.4 Extension to Previous Research . . . 25

2.3 Embeddings . . . 26

2.3.1 Document-Level Embeddings . . . 26

2.3.2 Graph-Level Embeddings . . . 27

2.3.3 Embeddings Used in Recommender Systems . . . 28

2.3.4 Extension to Previous Research . . . 29

3 Publications **31**

3.1 Contributions to the Main Publications . . . 31

3.2 Recommending Sellers to Buyers in Virtual Marketplaces Leveraging Social Information . . . 34

3.2.1 Abstract . . . 35

3.2.2 Introduction 35
3.2.3 Related Work 38
3.2.4 Datasets 39
3.2.5 Experimental Setup 42
3.2.6 Results 47
3.2.7 Conclusions 51
3.3 Predicting Trading Interactions in an Online Marketplace Through Location-Based and Online Social Networks . . . 52
3.3.1 Abstract 53
3.3.2 Introduction 53
3.3.3 Background 55
3.3.4 Datasets 64
3.3.5 Methodology 71
3.3.6 Results 85
3.3.7 Summary and Discussion 91
3.3.8 Conclusions and Future Work 94
3.4 Tell Me What You Want: Embedding Narratives for Movie Recommendations 96
3.4.1 Abstract 98
3.4.2 Introduction 98
3.4.3 Related Work 101
3.4.4 Empirical Analysis of Narratives 103
3.4.5 Feature Importance 115
3.4.6 Conclusions and Future Work 124
3.5 Evaluating Narrative-Driven Movie Recommendations on Reddit 126
3.5.1 Abstract 127
3.5.2 Introduction 127
3.5.3 Related Work 130
3.5.4 Reddit Narratives Evaluation Dataset 132
3.5.5 Experimental Setup 136
3.5.6 Results and Discussion 146
3.5.7 Conclusions and Future Work 153

4 Conclusions **155**
4.1 Results and Contributions 156

4.2 Implications 158
4.3 Limitations 160
4.4 Future Work 161

Bibliography **171**

1 Introduction

1.1 Motivation

Recommender systems are tools that help users to find and discover items of interest in large collections, such as books, movies, or people. In a common collaborative filtering scenario, a recommender system makes use of a user's history and predicts new items that user is likely to read, watch, or connect to. There exists a vast variety of studies about exploring new methods and algorithms to recommend items to users or users to users more efficiently [Barbieri et al., 2014; Backstrom and Leskovec, 2011]. However, not much attention has yet been paid to the usefulness of certain information available in external social networking sites for the task of recommending items or people to people. Especially in the context of online marketplaces and e-commerce websites, most of the current research still leverages information that is available within the platform, ignoring potentially useful social information. Further, the incorporation of narrative aspects originating from free text narratives, that explain current recommendation needs of users, into the recommendation calculation process is to a great extent yet unexplored.

Meanwhile, social information are a dime a dozen and social networks contain useful information about relations between users and their social characteristics [Coleman, 1988]. Our research community still lacks understanding of the value of social features from external sources, such as follower networks from Twitter[1] or common interests extracted from Facebook[2], for the application of e-commerce recommender systems. In the specific context-aware recommendation scenario called narrative-driven

[1] https://twitter.com
[2] https://facebook.com

recommendations [Bogers and Koolen, 2017], users have a specific idea what they are looking for, but the information need is often too complex to be articulated in the form of a few simple keywords. Users have built various discussion boards on the Web to ask peers for suggestions providing a descriptive narrative of their current recommendation needs. Although plenty of traditional recommender systems (e.g., a video game recommender on Steam[3]) exist that typically provide recommendations based on a user's profile or the content of a currently viewed item, they lack to accurately calculate recommendations based on narratives (e.g., "*Open world games, such as 'Grand Theft Auto V', that plays on a fictional planet*"). It is yet unexplored to what extent features extracted from narratives improve the recommender performance of default state-of-the-art recommender approaches. Specifically, our research community lacks understanding of the impact of positive (e.g., "*with happy ending*") vs. negative (e.g., "*without happy ending*") features extracted from narratives, and what aspects of such narratives are essential for the given recommendation task.

In Section 1.2 I provide a description of the problem statements, objectives, and approaches of this thesis. Section 1.3 shows the research questions (RQs) this thesis raises, followed by Section 1.4 listing the main publications this thesis builds upon. Section 1.5 summarizes the contributions and implications of this thesis, and finally this introduction concludes with an overview of the structure of the remainder of this thesis in Section 1.6.

1.2 Problem Statements, Objectives, and General Approaches

Problem Statements. The first part of this thesis deals with the problem of aiming and understanding the usefulness of social signals for recommendations on e-commerce websites. The focus is upon social signals that are typically available in online social network sites, such as Facebook. Particularly, this thesis should broaden our understanding of the usefulness

[3] https://store.steampowered.com

of social information, such as likes, comments, group joins, interest statements, or geographic check-ins, and corresponding similarity features for the task of recommending sellers to buyers in online marketplaces. The social information that is of particular interest in this work is categorized into topological including path-based features as well as homophilic features. Eventually, the first problem addressed in this thesis is a specific kind of link prediction problem—relevant for both virtual as well as physical social and recommender networks.

The fact, that there exist online communities using discussion boards for item suggestions, and traditional recommender systems struggle to produce suitable suggestions for such scenarios, is my motivation to check whether information extracted from narrative recommendation requests can be used to generate accurate algorithmic suggestions. On online discussion boards, users formulate their current recommendation needs in natural language texts to describe what they are looking for. Box 1.1 shows an example of such a narrative request and corresponding community suggestions on reddit in the domain of movies. The existence of such boards implies that the requirements of users exceed the possibilities of, for example, a recommender system simply based on filtering. Reasons why recommender systems struggle to solve this narrative-driven recommendation problem are manifold. The first obstacle is to understand text written by humans. This natural language processing problem is hard to tackle because of multiple reasons, for example, extracting the gist (i.e., keywords or key phrases) of the text, understanding text that is written in dialect, detecting typos, or being able to correctly interpret abbreviations (e.g., in the domain of movies "*LOTR*" is frequently used as "*The Lord of the Rings*" trilogy). The second issue for recommender systems is to incorporate the gist of the narrative in the recommendation calculation process. The challenge in this part is to correctly exploit mentioned entities (e.g., with "*James Bond*" a user could refer to a real person or the fictional character, or to the movies about him) and to incorporate the correct meaning of the extracted gist of the narrative (e.g., "*with happy ending*" requires suggested movies to have a plot with success for positive characters). The third obstacle is the missing insights into user preferences in narratives, and, particularly, whether users tend to illustrate their needs through (i) examples, (ii) by describing

SUBMISSION

"**[Request] Movies about writing/writers.**
Two of my favourites are *Secret Window* and *Stranger Than Fiction*. I also liked *The Ghost Writer*. [...] I'm not a fan of horror. I know there are probably a lot of 'inspirational' movies about writing out there (I vaguely recall one with Sean Connery?). [...]"

COMMENTS

"Adaptation."

"Sean Connery movie was Finding Forrester."

⋮

Box 1.1: **Request and Suggestions Example.** The narrative request of this reddit submission[4] contains three positive movies (i.e., *Secret Window*, *Stranger Than Fiction*, *The Ghost Writer*), a negative genre (i.e., *horror*), several positive keywords (i.e., *writing*, *writers*, *inspirational*), and a positive actor (i.e., *Sean Connery*). By commenting on this request the reddit community suggested the movies *Adaptation* and *Finding Forrester*.

the characteristics of desired items, or (iii) by a specific combination of both examples and characteristics. Also, the question whether positively associated aspects (e.g., *writing*, *writers*, *Secret Window*, and *Stranger Than Fiction*, in Box 1.1) are more important for calculating recommendations than negatively associated aspects (e.g., *horror* in Box 1.1) is still unanswered in the recent research on this topic. A better understanding of such preferences will have strong practical implications for improving the quality of algorithmic recommendations.

In summary, incorporating additional social and narrative features in the recommendation calculation process, represents scientifically interesting problems with lots of real-world applications.

Objectives. This thesis aims at providing insights into the usefulness of social signals that allow administrators of e-commerce websites to implement systems that provide suitable recommendations. One goal of this thesis is to extend existing analyses in two directions. First, I

[4] https://www.reddit.com/r/MovieSuggestions/comments/ssuhu

focus on a combined feature-based analysis of three different network sources, (i) an online social network, (ii) a location-based social network, and (iii) a trading network. Second, in contrast to, for example, Guo et al. [2011] or Zhang and Pennacchiotti [2013], I specifically focus on the impact of different topological and homophilic features, such as the common neighbors or the rooted PageRank in a network, common groups joins, or stated interests of users, since they provide actionable insights that can be used for decision making later. One main objective of this thesis is to evaluate to what extent it is possible to predict who will buy from whom and who will sell to whom, or in other words, who will trade with whom in the future. For this purpose, this thesis draws on different data sources, internal (i.e., the trading platform itself) as well as external networks. The focus is upon feature engineering and evaluation of various supervised and unsupervised methods as well as similarity metrics.

Another objective of this thesis is to investigate the usefulness of narrative features for recommender systems by systematically analyzing narrative-driven recommendations, and hence gain better insights into this novel topic. To that end, this thesis aims to quantify the difficulty of the narrative-driven recommendation problem by empirically analyzing a dataset that consists of narrative movie recommendation requests and corresponding suggestions. This work strives to evaluate the effects of positive vs. negative features in narratives, such as stated movie titles, genres, or keywords, that describe the gist of such recommendation requests. Further, this thesis analyzes the suitability of standard recommender algorithms as well as document and graph embedding techniques for supporting narrative features.

General Approaches. The first approach in this thesis to evaluate the utility of social features for e-commerce recommender systems is based on three different network sources, considering individual as well as combined usage. I evaluate different constellations of topological and homophilic features with several supervised and unsupervised learning methods. The second approach of this thesis to learn more about narrative-driven recommendations in general and to evaluate the utility of narrative features is an empirical analysis of a dataset about movie suggestions based on narratives, and the evaluation of state-of-the-art recommender algorithms for such a specific recommendation scenario.

1.3 Research Questions

This thesis captures studies about trading interaction recommendations based on the combination of social information from several networks, and the empirical analysis of narrative-driven recommendations as well as the evaluation of narrative features for recommender systems. In the first part, I analyze whether and to what extent trading interactions can be predicted based on social information originating from different networks (see RQ 1). In the second part, I investigate and quantify the difficulty of the narrative-driven recommendation problem and evaluate several state-of-the-art recommender algorithms to support narrative features for the computation of movie recommendations through a recommender network (see RQ 2).

RQ 1: How can we utilize social network features in recommender systems?

Problem. A growing body of research about utilizing social data for the task of recommending certain types of entities to people exists [Delporte et al., 2013; Jamali and Ester, 2010; Ma et al., 2011; Zhang and Pennacchiotti, 2013]. The majority of these studies focuses on exploring new methods and content- as well as collaborative-based algorithms to recommend items to people more efficiently. However, not much attention has yet been paid to the usefulness of certain social information available in social networking sites for the task of recommending items or people to people. Especially in the context of e-commerce websites and online marketplaces, the value of social features available in internal as well as external social networking platforms is to a great extent yet unexplored. Particularly, leveraging merged social information originating from several network sources for such a recommender task is rare. The aspect, that online marketplaces are typically dynamic with new users appearing every day, leads to another unexplored issue in such recommendation scenarios, which is the evolution of features and corresponding approaches over time.

Approach. To tackle this research question, in Eberhard and Trattner [2016] and Eberhard et al. [2019a] colleagues and I present a research effort that aims at recommending trading interactions between users from four different perspectives, (i) an online social, (ii) a location-based social, (iii) a trading network, (iv) and different combinations of them. To show if and to what extent it is possible to predict who will buy from whom and who will sell to whom, or in other words, who will trade with whom in the future, we consider both unsupervised as well as supervised methods applying different constellations of social features generated by the set of networks. Further, to investigate the feature performances over time, for any user visiting an online marketplace at a certain time we consider new sellers to this user at the that point of time that the user will most probably buy from in the future.

Findings and Contributions. In Eberhard et al. [2019a] colleagues and I find that features exctracted from the online social network and the location-based network are to some extend useful and achieve suitable results, whereas we achieve the best trading prediction results employing features from the trading network. It could be said that online and location-based social network information on their own or in combination achieve trading interaction prediction performance at an acceptable scale, which is convenient in specific settings (e.g., for cold-start predictions or in case that trading information is not available). The trading network information improves prediction performance strongly and does not necessarily require the addition of further information from other network sources. We further find that topological features are more suitable than homophilic features for the recommendation of trading interactions. This means that the information exploited from the network structure is more useful than other user related attributes represented through homophilic features. Further, the results in Eberhard and Trattner [2016] show that the performance of features varies over time, nevertheless most of the investigated features outperform the baseline at each point in time. We contribute to this research gap by providing recommender system designers and administrators of e-commerce websites with a guideline on what type of features and methods to apply in which use case to optimize the recommendation results.

RQ 2: How can we utilize narrative features in recommender systems?

Problem. Search engines are used to retrieve information when users can specifically articulate with a few simple keywords what they are looking for. When users vaguely know what they want but can not articulate it, they rely on recommender systems to explore large collections of items and find interesting items. In contrast, in narrative-driven scenarios users employ online boards and ask the community for item suggestions providing a free text narrative that explains their current recommendation needs. Other users then respond with relevant suggestions. The facts, that such online forums exist and that there is a lack of recommender systems that are able to accurately handle such narrative requests, are my motivation to automate the process of calculating suitable narrative-driven recommendations. The first step in this direction is to broaden our understanding of the potential causes for the hardness of the narrative-driven recommendation problem. We miss insights into user preferences in narratives, how users tend to illustrate their needs, and whether positively associated aspects are more important for calculating recommendations than negatively associated aspects. A better understanding of these narrative features will have strong practical implications for generating accurate algorithmic recommendations. The suitability of traditional recommender algorithms for such a narrative-driven scenario has—at least in the domain of movies—not been investigated until today.

Approach. To that end, in Eberhard et al. [2020] colleagues and I set out to learn more about narratives by analyzing a movie suggestion board from reddit (subreddit r/MovieSuggestions[5]). On this board, users typically describe movies they would like to see by composing a narrative in natural language including key aspects, suggested movies should satisfy. We present the first detailed empirical study of such narrative requests

[5] https://www.reddit.com/r/MovieSuggestions

and corresponding movie suggestions from the reddit community. To quantify the difficulty of this problem we analyze the diversity of requests and their corresponding suggestions. We evaluate the effects of positive vs. negative aspects extracted from narratives on reddit by individually investigating the feature correlations with the community suggestions. Further, we implement a narrative-driven movie recommender framework with several state-of-the-art recommender approaches and apply post-filtering and re-ranking strategies to refine the computed recommendations. For evaluation we measure the overlap between human suggestions from the reddit community and our purely algorithmic recommendations. We aim to identify the most essential narrative features for generating—in this case—algorithmic narrative-driven movie recommendations.

Findings and Contributions. The results presented in Eberhard et al. [2020] indicate that users mainly focus their requests on positive aspects, such as movies they liked or keywords that describe the movies they would like to see. The fact that there is a significant correlation between positive movies and suggestions also indicates that users frequently describe similar requests with differing movie examples. Additionally, we find that community suggestions are oftentimes more diverse than requests, meaning that highly similar requests are frequently answered with highly diverse movie suggestions by the reddit community. This fact indicates that implementing automatic narrative-driven movie recommendations is a hard problem. Initial results in Eberhard et al. [2019b] suggest that traditional recommender algorithms exhibit great potential for improvement when presented with a narrative, as they lack the proper means to include a priori specified requirements in the recommendation process. Further, we demonstrate that we can improve all implemented recommender approaches by applying post-filtering and re-ranking strategies using metadata available in the narratives of the initial requests on reddit. Additionally, the results of our recommender experiment in Eberhard et al. [2020] indicate that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power for narrative-driven movie recommendations on reddit. I strongly believe that the presented experiments will help researchers and practitioners to develop new and improve existing recommender approaches to better tackle the problem of narrative-driven recommendations, which also represents a fundamental problem in need of novel solutions for the advance of chat- and voicebots.

1.4 Main Publications

This cumulative thesis consists of the following four publications:

Article 1: [Eberhard and Trattner, 2016] Eberhard, L. and Trattner, C. (2016). Recommending Sellers to Buyers in Virtual Marketplaces Leveraging Social Information. *25th International Conference Companion on World Wide Web*

Article 2: [Eberhard et al., 2019a] Eberhard, L., Trattner, C., and Atzmueller, M. (2019). Predicting Trading Interactions in an Online Marketplace Through Location-Based and Online Social Networks. *Information Retrieval Journal*

Article 3: [Eberhard et al., 2019b] Eberhard, L., Walk, S., Posch, L., and Helic, D. (2019). Evaluating Narrative-Driven Movie Recommendations on Reddit. *24th International Conference on Intelligent User Interfaces*

Article 4: [Eberhard et al., 2020] Eberhard, L., Walk, S., and Helic, D. (2020). Tell Me What You Want: Embedding Narratives for Movie Recommendations. *31st ACM Conference on Hypertext and Social Media*

1.5 Contributions and Implications

This thesis features contributions on the topic of social and narrative aspects in recommender systems. Particularly, the main contributions of this thesis are as follows:

- First, with the analysis aimed at recommending trading interactions in an online marketplace by utilizing social network information from internal as well as external sources, this thesis gains insights into the usefulness of social features in recommender systems. This allows designers or administrators of e-commerce platforms to implement systems that provide suitable recommendations.

- Second, this thesis contributes to the narrative-driven recommendation problem by compiling, publishing, and empirically analyzing a reference dataset based on movie suggestion requests on reddit. Making this crowdsourced dataset available online, enables researchers to conduct independent analyses, advancing the state of research in the context of narrative-driven recommendations. With the in-depth investigation of the dataset, this thesis provides insights into user preferences in narratives, how users tend to illustrate their needs, and whether positively associated aspects are more important for calculating recommendations than negatively associated aspects.
- Third, with the evaluation of state-of-the-art recommender algorithms to support narrative features for the computation of movie recommendations, this thesis contributes to the novel research field of narrative-driven recommendations. The findings of this thesis serve as a stepping stone and support system designers to implement systems and even optimize state-of-the-art recommender algorithms to compute accurate narrative-driven suggestions.

Overall, the results and findings of this thesis will help researchers and practitioners to develop new and improve existing recommender systems to better incorporate social features in e-commerce applications and tackle the problem of narrative-driven recommendations. This also represents a fundamental problem in need of novel solutions for the advance of applications, such as chat- or voicebots.

1.6 Structure of this Thesis

In the remainder of this thesis I begin with reviewing most relevant work in Chapter 2. First, I position this thesis against other studies about link prediction for recommender systems and focus on commonly used features and methods in Section 2.1. Second, in Section 2.2 I focus on research about recommender networks including social information in such networks, context-aware recommender systems, and eventually narrative-driven recommender systems. Third, in Section 2.3 I discuss related work in the space of research on embeddings and their applications in recommender systems.

In Chapter 3 I present the main publications which combine to form this cumulative thesis as described in Section 1.4. Particularly, I outline my personal contributions to each of these publications in Section 3.1. Table 1.1 provides an overview of the main publications, the tackled research questions in this thesis, the related topics, as well as the corresponding main contributions for each publication.

To conclude this thesis in Chapter 4, I provide a summary of the main findings and contributions in Section 4.1, I briefly discuss potential implications and limitations of this work in Sections 4.2 and 4.3, and outline future work in Section 4.4.

Table 1.1: **Overview of the Main Publications.** This table gives an overview of the publications of this thesis and their corresponding research questions, topics, and main contributions.

Article	**RQ**	**Topic**	**Main Contribution**
Article 1 [Eberhard and Trattner, 2016]	RQ 1	Recommending trading interactions	Recommending trading interactions in an online marketplace by utilizing social network features and their usefulness over time.
Article 2 [Eberhard et al., 2019a]	RQ 1	Recommending trading interactions	Recommending trading interactions in an online marketplace by utilizing social network information from internal as well as external sources. Investigating different networks individually as well as combined to evaluate their trading prediction capabilities with the aid of a rich set of topological and homophilic features.
Article 3 [Eberhard et al., 2019b]	RQ 2	Evaluating and optimizing narrative-driven recommendations	Compiling, publishing, and empirically analyzing a reference dataset based on narrative movie suggestion requests on reddit.
Article 4 [Eberhard et al., 2020]	RQ 2	Analyzing and evaluating narrative-driven recommendations	Evaluating and optimizing state-of-the-art recommender algorithms to support narratives for the computation of movie recommendations through a recommender network.

2 Related Work

This chapter provides an overview of all topics related to the studies presented in this thesis. The first part is link prediction which I discuss in Section 2.1. It covers the link prediction methods supervised and unsupervised learning and the link prediction feature types topological and homophilic features. In Section 2.2, I review literature related to recommender systems that use social information, context-aware recommender systems, and narrative-driven recommender systems, which forms the second part. Finally, in Section 2.3, I review the third part which is about embeddings in general, document- and graph-level embeddings, and embeddings used in recommender systems.

2.1 Link Prediction

Link prediction is a well-established method for predicting new or missing edges in networks. In social networks, the prediction of edges between nodes representing, for example, interactions or relations between persons is a challenging task. In 2003, Liben-Nowell and Kleinberg [2003] presented a comprehensive fundamental analysis. They defined the link prediction problem as the task of predicting edges that will be added to a social network at a given snapshot during a given interval. Predicting such edges is used to suggest promising interactions between persons in a social network. The authors investigated a co-authorship network among scientists. They evaluated their intuition that scientists who have never written an article together will collaborate in the future more likely, if they are close in the network, meaning that they have common colleagues and travel in similar circles. Their results indicate that the topology of the network contains latent information that can be exploited to predict future interactions. Link prediction is further applied to, for example, recommend friends, suggest business associates, or forecast whether a user will buy a specific product in an online store [Guo et al., 2011; Zhang et al., 2013; Barbieri et al., 2014].

Typically, the link prediction literature refers to two types of learning approaches, unsupervised learning and supervised learning, which I discuss in the following section.

2.1.1 Link Prediction Methods

Supervised Learning. Supervised link prediction is employed in various machine learning methods. It is commonly used to predict edges between nodes in a network whenever label information is available. It treats the link prediction task as binary classification problem using a set of node pairs that are actually linked as positive class and a set of node pairs that are not connected as negative class. These two classes are used to train the classifier that is going to be applied. After the training phase, the classifier is used to predict whether two arbitrary nodes are likely going to be connected or not [da Silva Soares and Prudêncio, 2012].

Hasan et al. [2006] published a work about supervised link prediction on a co-authorship network. Similarly to Liben-Nowell and Kleinberg [2003], the authors considered a social network with interactions as edges representing the co-authoring of research articles. They used at least author information and publication year of each article as data for their link prediction experiment. First, they split the set of articles based on the publication years chronologically into training and test set. Then, for classification they used author pairs that already existed in the training set, but did not publish any articles together in this period. Each of these pairs represents either a positive or a negative example, depending on whether the two authors collaborated in the test set period or not. They applied several classification algorithms with the target to predict edges between authors who collaborated in the test set period by successfully distinguishing between the positive and negative class. Consequently, the authors solved this binary classification problem by supervised learning mainly employing effective features from the topology of the network.

There exists a vast variety of classification algorithms for supervised link prediction tasks. In their work, Hasan et al. [2006] evaluated the algorithms *bagging*, *decision tree*, *k-nearest neighbors*, *multilayer perceptron*, *naïve Bayes*, *radial basis function network*, and *support-vector machine*. Basically, their performances are comparable but depending on case of application some usually work better than others. Further algorithms commonly employed for supervised link prediction yielding promising results are, for example, *AdaBoost*, *logistic regression*, or *random forest* [Cranshaw et al., 2010; Fire et al., 2011; Backstrom and Leskovec, 2011; Leskovec et al., 2010; Steurer and Trattner, 2013c].

However, to better estimate the performance of supervised learning algorithms it is suggested to compare the results with unsupervised link prediction approaches, such as *collaborative filtering* [Steurer and Trattner, 2013a; Bischoff, 2012].

Unsupervised Learning. Unsupervised link prediction does not require labeled training information. Similarity scores between nodes in a network are computed, typically based on node neighborhoods, path information, or other similarity characteristics between nodes. A higher score implies a higher similarity between two nodes. Then, all pairs of nodes without edges are ranked in descending order according to their scores, having pairs with the highest scores on top of the list. The top-N pairs of nodes without edges between them are then predicted to be linked. Unsupervised algorithms are evaluated by (randomly) dividing the set of edges of the network into a training set, treated as known information, and a test set. Metrics such as precision, recall, or F1 score are commonly used to quantify the prediction accuracy. This method is simple to implement but exhibits some limitations, such as finding the optimal threshold N for the number of pairs to be predicted or the difficulty of combining information for the prediction [da Silva Soares and Prudêncio, 2012; Lü and Zhou, 2010].

2.1.2 Link Prediction Features

To assess similarities between users in a social network, important information about users, such as common interests, or information extracted from the topology of the network, such as common neighbors, is typically used. The computed similarities are then employed as features for the applied link prediction task. Generally, link prediction features can be divided into two types, (i) *topological features* based on the topology or proximity of nodes of the network, and (ii) *homophilic features* based on homophily which describes the tendency for friend- or relationships to occur between individuals [Thelwall, 2009; Steurer and Trattner, 2013a; Coleman, 1988].

Topological Features. If the structure of a network is known topological features can be applied in order to determine the similarities between nodes. Measures such as the number of neighbors two users have in common [Liben-Nowell and Kleinberg, 2007], the Jaccard's coefficient of the neighborhood of two users (i.e., number of common divided by number of total neighbors) [Salton and McGill, 1983], the Adamic Adar with respect to the node degrees of the common neighbors of two users [Adamic and Adar, 2003], or the preferential attachment score as the multiplication of the numbers of neighbors of two users [Barabasi and Albert, 1999; Newman, 2001] are typically used as topological features for link prediction tasks. Depending on case of application they often lead to valuable results. Steurer and Trattner [2013a] employed more detailed topological features. For their directed network the authors distinguished between incoming and outgoing edges, the consequence being that they obtained refined topological features, each of which split into incoming and outgoing feature. Further topological features for directed networks are the so-called transitive friends feature, which is the number of outgoing neighbors of a user intersected by the number of incoming neighbors of another user, and the opposite direction friends feature measuring the reciprocity between two users [Fire et al., 2013].

As an extension of topological features which focus on the local neighborhood of a node, *path-based features* consider richer connectivity information. For example, the Katz measure [Katz, 1953] also takes nodes beyond the local neighborhood of a node into account. It measures the strength of the connection between two arbitrary nodes by considering the number of different paths between these nodes and the path lengths. The more paths two nodes are connected with and the shorter these paths are, the stronger is the connection between these nodes. A damping factor steers the importance of shorter vs. longer paths. A high weight on short paths yields prediction results similar to the common neighbors feature, because paths of length three or more contribute little to the final aggregation [Liben-Nowell and Kleinberg, 2003]. Another established path-based feature is the rooted PageRank [Liben-Nowell and Kleinberg, 2003]. As an adaption of the PageRank algorithm [Page et al., 1999], it provides the stationary probability distribution sets of nodes regarding a specific starting node, which is used as ranking for link prediction tasks.

Homophilic Features. Thelwall [2009] described homophily as the principle that an interaction or relation between people rather occurs if they are similar than between dissimilar people. In social networks, the target of homophily is to perceive and localize behavioral, cultural, genetic, or material information that flows through a network. It structures the edges of a network of every type or relationship, such as marriage, friendship, information transfer, or work advice. For the personal environment common homophilic attributes are age, religion, education, occupation, and gender. Homophilic attributes are crucial for the user behavior, the information users receive, and the attitudes they form, as investigated by McPherson et al. [2001] in the context of social networks. To predict partnerships in social networks Steurer and Trattner [2013a] used attributes such as groups, interests, interactions, events, and regions. To derive homophilic features from these attributes they computed measures such as common items (i.e., number of items of an attribute two users have in common), total items (i.e., number of total items of an attribute of two users), Jaccard's coefficient of items (i.e., number of common divided by number of total items of an attribute of two users), or cosine similarity of item vectors.

Overall, link prediction tasks can be approached using different feature types depending on their availability, both individually as well as in combination covering single and even multiple networks. Cranshaw et al.

[2010] applied a hybrid approach combining a location-based network with an online social network. They used the location-sharing Facebook application called Locaccino and tried to predict links in the online social network. Steurer and Trattner [2013a] also combined online social network data with location-based social network data in their partnership prediction experiments. In the area of predicting trading interactions, Guo et al. [2011] investigated usefulness of social networks and several different features to predict seller-buyer interactions. They computed centrality metrics (e.g., PageRank) as well as homophilic metrics (e.g., number of friends a seller and a buyer have in common) to predict trading interactions.

2.1.3 Extension to Previous Research

The literature discussed above shows that link prediction is a well-studied research topic. However, little work using multiple networks of different types and, in particular, their combination for predicting links has been conducted. Especially studies in the area of trading interactions between sellers and buyers are rare. In this thesis, I use similar features as Guo et al. [2011], but rely not solely on social network data and determine whether there is also valuable information in people's location positions for predicting trading interactions. I investigate location-based network data to understand whether features, such as the number of times sellers and buyers have been in the same location, bears a signal that can be exploited to better predict links between sellers and buyers in the future. Particularly, I implement a rich set of topological features, such as common neighbors or the path-based rooted PageRank, and homophilic features, such as number of common groups or interests, for three different network sources. Besides the evaluation of each individual feature I determine the performance of different combinations of features and combinations of network sources. In addition to this, for the prediction experiments I apply supervised and unsupervised learning approaches and further determine the extent to which the induced features and similarity metrics are useful over time.

2.2 Recommender Networks

Recommender systems are tools and techniques to provide suggestions for items to be of interest to a user [Ricci et al., 2011]. The target of link prediction tasks is often to recommend friendships, interactions, or other relations between users in a network making the topic of link prediction related to recommender systems [Huang et al., 2005; Scellato et al., 2011; Lü and Zhou, 2011].

Newman [2018] describes recommender networks as the representation of users' preferences for items, such as products of an online store. Similar to supermarket chains which collect purchase data from their regular customers (e.g., by a customer card with a barcode), online stores record which customers buy which products. Additionally, they sometimes collect information about whether the customers like the bought products or not by a rating system. Basically, a recommender network is represented as bipartite network, which consists of two types of nodes, one representing items or products and the other representing users. The edges in such a network between users and items show whether a user bought or liked an item. Optional edge weights indicate how often a user bought a specific item or how much they liked it [Newman, 2018]. Traditional research in recommender networks focuses on algorithmic advantages in common scenarios, such as applying users' histories and profiles to compute recommendations [Christakou et al., 2007; Ghosh et al., 1999; Mak et al., 2003; Perny and Zucker, 2001].

Adomavicius and Tuzhilin [2005] distinguish between content-based, collaborative, and hybrid approaches for calculating recommendations. Content-based approaches, such as term frequency–inverse document frequency (TF–IDF) or decision trees, rely on similarities between items based on item descriptions. Collaborative approaches filter for information or patterns using collaboration among multiple data sources [Terveen and Hill, 2001]. To make predictions about the interests of a user they employ preferences of other users. The basic idea behind these approaches is that if a user has the same preference (e.g., rating) on an item as another user, it is more likely that this user has a similar preference as the other user than a random user on another item [Adomavicius and Tuzhilin, 2005]. In contrast to such user-based algorithms, item-based collaborative filtering measures similarities between items based on user ratings for recommendation. Table 2.1 illustrates item-item similarity by an example ratings matrix. To compute recommendations for an item, similarities to all other items are computed with the most similar items to be recommended. In this example, *6* is the most similar item to item *4* since the users A and C rated both items low. There are several different ways to compute such item-item similarity. Two common methods are correlation-based and cosine-based similarity. Cosine-based similarity between two items is measured by using their columns in the ratings matrix as vectors and computing the cosine of the angle between them. Correlation-based similarity between two items is measured by computing the Pearson correlation of their columns isolating all co-rated cases (i.e., cases where a user rated both items) [Sarwar et al., 2001]. Matrix factorization is a specific collaborative filtering approach with the target to decompose the user-item interaction matrix into two separate matrices. As shown in Figure 2.1, with this method it is possible to generate interaction predictions for every user-item combination by computing the product of the two matrices [Koren, 2008].

Table 2.1: **Item-Item Similarity.** This example user-item ratings matrix demonstrates the item-item similarity, which is computed by looking into co-rated items only. In this case, item IV is the most similar one to item *VI* because users A and C rated both items low. Adapted from Sarwar et al. [2001].

User \ Item	***I***	***II***	***III***	***IV***	***V***	***VI***
A	**5**		**1**	**1**		**2**
B		**2**		**4**		**4**
C	**4**	**5**		**1**	**1**	**2**
D			**3**	**5**	**2**	
E	**2**		**1**		**4**	**4**

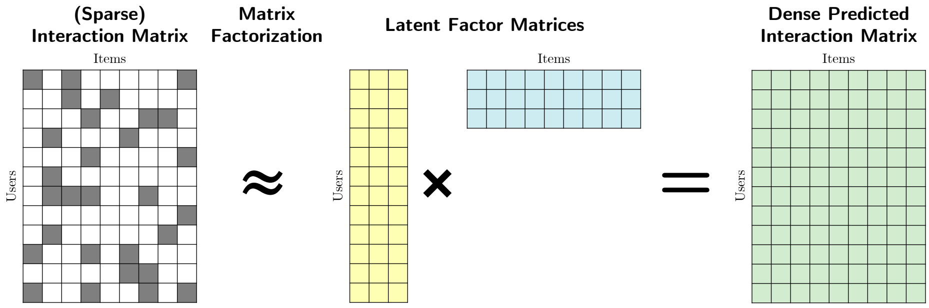


Figure 2.1: **Matrix Factorization.** This figure shows the basic principle of matrix factorization based on a ratings matrix. The target is to decompose the user-item ratings matrix into two separate matrices (i.e., a user and an item matrix that contain latent factors), so that by computing the product of these two matrices it is possible to generate rating predictions for every user-item combination. Adapted from Cai et al. [2018].

2.2.1 Social Information in Recommender Networks

In a variety of studies recommender systems have exploited social information to compute item or product recommendations. For example, Jamali and Ester [2010] used social relations between users to predict user ratings, Delporte et al. [2013] incorporated implicit feedback and data from a social graph to recommend items and places, or Lu et al. [2010a] proposed a framework to predict the quality of user reviews by exploiting social information about the identities and social connections of the review authors. Their evaluation shows that social information helps to increase the prediction accuracy of review quality. In the context of e-commerce in recommender networks, studies typically focus on algorithmic advances to predict the rating or ranking of items people might prefer. However, Zhang and Pennacchiotti [2013], for example, found that in a cold-start scenario, top-level categories on eBay can be better predicted by using social data from Facebook in terms of “likes”. Further, Ma et al. [2011] analyzed the social recommendation problem based on matrix factorization evaluated on e-commerce data. They could improve their results by adding social information as regularization terms to their factorization approach.

2.2.2 Context-Aware Recommender Systems

Besides user profiles and histories, context-aware recommender systems employ contextual information to compute recommendations that are well suited to the current needs of the user. Contextual information is, for example, the time of the day, the location or interests of a user in a specific situation, or conditions which influence the user's decisions (e.g., weather or physical condition) [Hariri et al., 2013; Oku et al., 2006; Adomavicius et al., 2005]. Many previous recommender studies have shown potential for relevant contextual information when providing recommendations [Lamprecht et al., 2015; Adomavicius and Tuzhilin, 2011]. Such contextual information can be obtained in different ways. The first one is *explicitly*, meaning that information is gathered by directly asking users questions or collecting information by other means. For example, an online platform may obtain contextual information by asking users to fill out a form with specific questions or by exploiting text of a user's profile. The second way is *implicitly*, meaning that the information is not directly entered by the user, but can be obtained from the data or the environment (e.g., location change of a user, timestamp of a transaction, or metadata of an item that is mentioned by a user). To that end, pre- and post-filtering techniques can be applied to capture relevant context during the recommendation process. These methods select a relevant set of data and filter out irrelevant recommendations or adjust the ranking of the obtained recommendation list based on a given context [Adomavicius and Tuzhilin, 2011].

2.2.3 Narrative-Driven Recommender Systems

A specific context-aware recommendation scenario called narrative-driven recommendation was presented by Bogers and Koolen [2017]. Besides past transactions of users, in such a scenario recommendations are computed based on a narrative description of the current needs and interests of a user. Narrative-driven recommendations are related to conversational-based recommender systems, where users ask for suggestions in a community and other users then come up with suggestions and possible explanations for their choices. When providing an appropriate narrative the user obtains items assimilated to the described situation. In contrast to traditional recommender systems focussing on user histories or profiles, this treatment enables users to specifically define their current recommendation needs depending on a specific situation. For example, in a movie recommendation scenario a user might focus one request on family movies while on another day the same user might be in the mood for watching a horror movie. Since narrative-driven recommender systems mainly focus on the provided description, these two requests are considered independently although originating from the same user, meaning that the content of the two requests do not influence each other. In the domain of movie recommendations, Bogers [2015] investigated discussion threads from the IMDb message boards containing user requests for movies to watch. Besides content and metadata, such as movie descriptions, genres, or languages, the author found that searching for movies by describing their contents in a narrative way is essential for movie selection practices.

2.2.4 Extension to Previous Research

In contrast to previous work, I present the first in-depth analysis and evaluation of recommender algorithms to support narratives for the computation of movie recommendations through a recommender network. Using data from IMDb, I implement item-based collaborative filtering and matrix factorization based on user ratings, a network-based approach using casts and crews from movies, and a TF–IDF and a simple `doc2vec` approach both based on movie descriptions and user reviews. For evaluation, I use a crowdsourced dataset from reddit submissions providing movie recommendation requests from users in a narrative way and comments including the respective movie suggestions of the reddit community. The obtained results suggest to refine the implemented state-of-the-art algorithms by re-ranking the computed recommendations via post-filtering techniques based on the specific user requirements. In contrast to the study of Panniello et al. [2009] that constitutes a first step towards the comparison of pre- and post-filtering using just one contextual variable for each applied dataset, I introduce and combine several post-filters and evaluate their utility in the context of narrative-driven movie recommendations. This procedure leads to a substantial improvement in the quality of the recommendations.

2.3 Embeddings

2.3.1 Document-Level Embeddings

With the neural probabilistic language model `word2vec`, Mikolov et al. [2013a] proposed a method to learn high-quality embeddings for words in texts, where each word is mapped to a unique vector. `word2vec` has become a well-established model to provide state-of-the-art performance on linguistic tasks [Levy et al., 2015; Mikolov et al., 2013b; Mnih and Hinton, 2009]. It is based on the hypothesis that words that emerge in similar contexts incline to have similar meanings and are in further consequence located closely in the vector space. Since the learned word vectors capture linguistic regularities and patterns, with vector arithmetic operations analogy questions can be answered. For example, the resulting vector of the operation `vec("King") - vec("Man") + vec("Woman")` is closer to `vec("Queen")` than to any other word vector [Mikolov et al., 2013c], as shown in Figure 2.2. The idea of vector embeddings has been adapted successfully for various domains, for example, `community2vec` [Martin, 2017], `item2vec` [Barkan and Koenigstein, 2016], `prod2vec` [Grbovic et al., 2015], `emoji2vec` [Eisner et al., 2016], or `doc2vec` [Le and Mikolov, 2014].

`doc2vec` was proposed by Le and Mikolov [2014] as an enhancement of `word2vec` to allow the learning of document-level embeddings. It introduces numeric vectors each of them representing a paragraph of a document. Such a vector acts as a memory that remembers the topic of the respective paragraph. This unsupervised learning algorithm learns fixed length feature representations for arbitrary lengths of texts. The authors provide two different approaches within `doc2vec`, namely *distributed memory* (`PV-DM`) and *distributed bag of words* (`PV-DBOW`). In case of `PV-DM`, the authors apply the inspiration of `word2vec`, where the embedded vectors are used for a prediction task about the next word in a context. `PV-DM` averages or concatenates the paragraph vector and the word vectors to predict

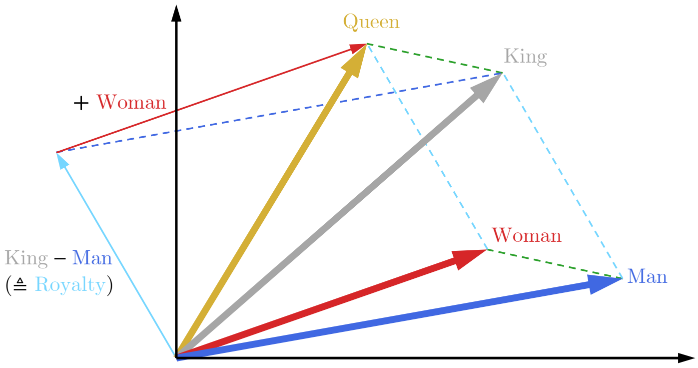


Figure 2.2: **Arithmetic Operations with Word Embeddings.** Analogy questions can be answered with vector arithmetic operations. The 2D vector representation example in this figure highlights that the operation `vec("King") - vec("Man") + vec("Woman")` leads to `vec("Queen")`. Similar semantic relations—in this case the word *Royalty* (represented through `vec("Royalty")`) as relation between *Man* and *King* as well as *Woman* and *Queen*—facilitate such simple arithmetic vector operations. Adapted from Mikolov et al. [2013c].

the next word in a sentence. `PV-DBOW` works similar to the `skip-gram` model [Mikolov et al., 2013b] except that the word vectors are replaced by paragraph vectors. In this version the order of words in the document is ignored.

2.3.2 Graph-Level Embeddings

Based on `word2vec`, Perozzi et al. [2014] introduced an embedding approach called `DeepWalk` that employs the co-occurrences of nodes in random walk paths in a graph. The nodes in recorded walking paths can be compared to words in sentences in a text corpus. In 2016, Grover and Leskovec [2016] proposed the embedding technique `node2vec` that is also based on random walks. As an extension to `DeepWalk`, `node2vec` diversifies the neighborhood of a node by utilizing fixed-length random walks with a mixture of breadth-first search and depth-first search schemes. The random walks are controlled by two parameters that steer how fast a walk spreads and how fast a walk leaves the neighborhood of the starting node [Chen et al., 2019].

2.3.3 Embeddings Used in Recommender Systems

In the large and well-investigated research field of recommender systems and algorithms, there exists a vast variety of studies based on `word2vec` and its extensions `doc2vec` and `node2vec` partly exhibiting outstanding performances [Musto et al., 2015; Ozsoy, 2016; Stiebellehner et al., 2018].

Elsafty et al. [2018] showed in their job posting recommendation scenario that `doc2vec` outperforms not only `word2vec` but also the well-established content-based recommender approach TF–IDF when using job titles combined with full-text job descriptions.

Manotumruksa et al. [2016] used the `skip-gram` model to learn word embeddings on venue and user data from Foursquare to provide context-aware venue recommendations. In the learning to rank system, they computed cosine similarities between venue and user vectors and used them as feature. Their results show the word embedding approach outperforming approaches without embeddings.

Another work in this context was done by Ozsoy [2016]. The authors used `word2vec` to train embeddings representing item and user profiles for venue recommendations to make predictions on where users will check-in next. The authors showed that their word embedding approach can obtain state-of-the-art results comparable to other content-based recommender approaches or matrix factorization.

Musto et al. [2015] obtained similar results in their empirical evaluation about movie and book recommendations. They compared several word embedding approaches exploiting textual information from Wikipedia against user-to-user and item-to-item collaborative filtering and matrix factorization. They conclude that embedding techniques yield promising results for the recommendation task.

For calculating mobile app recommendations, Stiebellehner et al. [2018] applied a `doc2vec` representation of users and items. The authors used app usage histories and textual app descriptions to learn vectors representing mobile app users. Further, they trained embeddings on additional user and app metadata. The quality of recommendations provided by their `doc2vec` approach was remarkably higher than by other state-of-the-art algorithms.

Chen et al. [2017] introduced a spectral clustering-based collaborative filtering recommender framework based on `node2vec`. The authors used a bipartite user-item network from a real-world dataset for their experiments and obtained results that exhibit positive effect on the improvement of baseline algorithms. In their followup work [Chen et al., 2019], they extended their approach by incorporating category information and combining multiple bipartite networks to even further improve the performance of their recommender framework.

Musto et al. [2019] proposed a semantics-aware recommendation strategy that uses graph-level embeddings. The authors employed users, items, and entities gathered from DBpedia to create a tripartite graph that served as basis for graph-level embeddings. Their comparison of different techniques for recommendation tasks shows that `node2vec` obtained the best results on all used datasets.

Kallumadi and Hsu [2018] evaluated the effectiveness of query-based interactive movie recommendations on IMDb data using graph-level embeddings. They created meta paths with different entities (e.g., users, movies, genres) to build movie networks as basis for their embeddings and obtained suitable results with `node2vec`.

2.3.4 Extension to Previous Research

Alongside the method of applying post-filtering techniques to improve the movie recommendation quality, I introduce an approach that incorporates key aspects of narratives through embeddings. To that end, I present the first in-depth empirical analysis of narratives from a movie suggestion board to quantify the difficulty of the narrative-driven recommendation problem. Further, I present the evaluation of embedding narratives through `doc2vec` based on movie descriptions and user reviews and `node2vec` based on different movie graphs. Getting rid of the demand of using post-filters steers to new opportunities for easily incorporating further types of entities or even larger parts of narratives.

3 Publications

3.1 Contributions to the Main Publications

This section lists all of my contributions to the main publications of this cumulative thesis.

Article 1: [Eberhard and Trattner, 2016] Eberhard, L. and Trattner, C. (2016). Recommending Sellers to Buyers in Virtual Marketplaces Leveraging Social Information. *25th International Conference Companion on World Wide Web*

As the primary author of this article, I was responsible for designing the approach, collecting, preprocessing, cleaning, and preparing the data, conducting and executing the experiments, and examining all results. To that end, I extended and utilized a framework that was originally written in Python by Michael Steurer and Christoph Trattner for executing the experiments.

The main ideas for this article, to understand to what extent certain social information and corresponding user-similarity features are useful for the task of recommending sellers to buyers, and to evaluate the feature performance over time, were proposed by Christoph Trattner. Both authors contributed to the interpretation of the results and the writing of the paper.

Article 2: [Eberhard et al., 2019a] Eberhard, L., Trattner, C., and Atzmueller, M. (2019). Predicting Trading Interactions in an Online Marketplace Through Location-Based and Online Social Networks. *Information Retrieval Journal*

For this follow-up article to the first one, as the main author I compiled the data, performed the experiments and produced all tabular and visual representations of the corresponding results.

The main idea, to evaluate the trading interaction predictability of different features and feature sets across multiple networks and their combinations, was proposed by Christoph Trattner. He was also responsible for setting up the study design and experiments. The decisions about the applied methods originated from discussions between all authors. The interpretation of the results and the writing of the paper was performed by all authors.

Article 3: [Eberhard et al., 2019b] Eberhard, L., Walk, S., Posch, L., and Helic, D. (2019). Evaluating Narrative-Driven Movie Recommendations on Reddit. *24th International Conference on Intelligent User Interfaces*

As the main author of this article, I was responsible for the design and implementation of the narrative-driven recommender framework, for the evaluation and optimization of different state-of-the-art recommender algorithms, as well as for producing all tabular and visual representations of the corresponding framework and results.

Simon Walk proposed the fundamental idea to this article, to automate the process of narrative-driven movie recommendations on reddit, which was refined in discussions between all authors. The design, implementation, and execution of the crowdsourcing experiment in this article was performed by Lisa Posch. All authors discussed and interpreted the results, and contributed to writing the paper.

Article 4: [Eberhard et al., 2020] Eberhard, L., Walk, S., and Helic, D. (2020). Tell Me What You Want: Embedding Narratives for Movie Recommendations. *31st ACM Conference on Hypertext and Social Media*

My contributions to this article were devising and formulating the different hypotheses, designing the approach, analyzing the data, executing the conducted experiments, as well as visually representing the obtained results. To that end, I extended our existing narrative-driven recommender framework to handle and process the new approaches based on document and graph embedding techniques.

Based on the findings from the previous work, Denis Helic proposed the idea for this article, to gain better understanding of the narrative-driven recommendation problem by empirically analyzing the crowdsourced dataset. The idea to additionally conduct a prediction experiment through different embedding approaches was stemmed from discussions among all authors. All authors of this article were involved in interpreting the results and writing the paper.

3.2 Recommending Sellers to Buyers in Virtual Marketplaces Leveraging Social Information

This article tackles the first research question by contributing to the sparse research field of social impact for the task of recommending sellers to buyers in virtual marketplaces. Specifically, my co-author and I were interested in social signals that could benefit the predictability of trading relations between sellers and buyers. We deal with the following problem: for any user visiting an online marketplace at a certain time we generate a list of new sellers that the user has not observed previously and will most probably buy from in the future. To that end, we introduce several social features and similarity metrics that we then use as input for a user-based k-nearest neighbor collaborative filtering method. Further, we evaluate to what extent these features and corresponding approaches are useful over time. Although this aspect is typically neglected in recommender systems research, we are interested in this kind of issue, since online marketplaces are usually very dynamic where new sellers and buyers appear nearly every day.

The results presented in this article indicate that most types of the applied social information and features are useful to tackle the defined research question. We find social information such as joined groups or stated interests more useful, while, for example, location-based social data, does not significantly improve the predictability of trading relations. Further, our results reveal that features significantly vary in their predictive power over time, while others show more stable behaviors. The findings of this article could support researchers interested in recommender systems, online marketplace administrators, as well as engineers interested in feature engineering.

3.2.1 Abstract

Social information such as stated interests or geographic check-ins in social networks has shown to be useful in many recommender tasks recently. Although many successful examples exist, not much attention has been put on exploring the extent to which social impact is useful for the task of recommending sellers to buyers in virtual marketplaces. To contribute to this sparse field of research we collected data of a marketplace and a social network in the virtual world of Second Life and introduced several social features and similarity metrics that we used as input for a user-based k-nearest neighbor collaborative filtering method. As our results reveal, most of the types of social information and features which we used are useful to tackle the problem we defined. Social information such as joined groups or stated interests are more useful, while others such as places users have been checking in, do not help much for recommending sellers to buyers. Furthermore, we find that some of the features significantly vary in their predictive power over time, while others show more stable behaviors. This research is relevant for researchers interested in recommender systems and online marketplace research as well as for engineers interested in feature engineering.

3.2.2 Introduction

Utilizing social data for the task of recommending certain types of entities to people has gained great popularity recently [Delporte et al., 2013; Jamali and Ester, 2010; Ma et al., 2011; Zhang and Pennacchiotti, 2013]. Although a growing body of research exists, exploring new methods and algorithms to recommend items to people more efficiently, not much attention has yet been paid to the usefulness of certain social information available in social networking sites for the task of recommending items or people to people. Especially in the context of e-commerce websites and online marketplaces, the value of social information available in external social networking platforms is to a great extent yet unexplored. Most of the current research still leverages information that is available within the e-commerce platform, ignoring useful social information [Lacic et al., 2015]. To contribute to this area of research, we present in this paper a work in progress of a research effort that aims at understanding the usefulness of social signals for recommendations in e-commerce websites. We focus upon social signals that are typically available in online social network sites such as Facebook. In particular, we are interested in understanding the usefulness of social information such as likes, comments, group joins, interest statements, geographic check-ins, and corresponding similarity features for the task of recommending sellers to buyers in online marketplaces. For the recommendation task we have chosen a user-based k-nearest neighbor collaborative filtering approach.

Problem Statement. In this paper we deal with the following problem: for any user visiting an online marketplace (for whom we also have social networking information available) at a certain time we generate a list of new sellers (sellers the potential user has not observed previously) that the user will most probably buy from in the future. To do so, we try to introduce several social features and similarity metrics from the social networking activities of the user. We use them to train a set of user-based k-nearest neighbor collaborative filtering models based on these features to generate an optimal list of top-N relevant new sellers to the buyer at a given time to investigate what types of features are the most useful ones at that point of time.

Research Questions. The following research questions were posed:

- *RQ 1: Recommending Sellers to Buyers.* Knowing that social networking information and corresponding features can help in recommending products to people in online marketplaces [Lacic et al., 2015], to what extent are certain social information and corresponding user-similarity features useful in a user-based k-nearest neighbor collaborative filtering setting for the task of recommending sellers to buyers?
- *RQ 2: Feature Performance over Time.* To what extent are these features and the corresponding collaborative filtering approaches useful over time? This question is typically neglected in recommender systems research, but one which we argue is important to ask, since online

marketplaces are typically very dynamic where new sellers and buyers appear nearly every day.

Results. Based on a number of experiments in the virtual world of Second Life we find that not all social information and corresponding similarity metrics are useful in a user-based collaborative filtering setting to recommend new sellers to buyers in the marketplace of Second Life. In fact we find that social information such as joined groups or stated interests induced from the online social network are almost as useful as historical information such as product categories directly induced from the marketplace. Interestingly, compared to a most popular baseline, location-based social information is not very suitable to tackle the defined problem. This is in line with previous observations that people in virtual worlds are not bound to certain places due to the possibility to teleport to places [Balby Marinho et al., 2015].

Contributions. The main contributions of this work are manyfold, but can be broken down to the following points:

- First, we believe this study is unique in a way that it tackles the problem of recommending new sellers to buyers in marketplaces.
- Second, the study contributes to a better understanding of the seller-buyer recommendation problem by investigating the extent to which social information is useful in a user-based collaborative filtering setting through a number of offline experiments—a feature that has not been investigated yet.
- Finally, the study shows to what extent the induced social features are useful over time—an important property that to the best of our knowledge has not been reported yet.

The paper is structured as follows: In Section 3.2.3 we provide an overview of relevant related work in this area. The datasets used in this work are described in Section 3.2.4. Section 3.2.5 provides a detailed description of the experimental setup. The results of our experiments are presented in Section 3.2.6. Finally, Section 3.2.7 reports some conclusions that can be drawn from this work and highlights some future directions which are worth to be further explored.

3.2.3 Related Work

Using social information to provide or improve recommenders is a relatively new strand of research. Most notable work in this direction has been performed recently in the context of, for instance, recommending points-of-interest to people (e.g., Macedo et al. [2015]), recommending tags to people (e.g., Feng and Wang [2012]), or predicting social interactions (e.g., Bischoff [2012], Steurer and Trattner [2013b]) or relations (e.g., Trattner and Steurer [2015]).

In the context of e-commerce not much work has been performed yet and only a few studies exist typically focusing on algorithmic advances to predict the rating or ranking of items people might prefer [Delporte et al., 2013; Jamali and Ester, 2010; Lacic et al., 2015; Ma et al., 2011; Zhang and Pennacchiotti, 2013]. Studies investigating the extent to which social information is useful for the task of recommending sellers to buyers are rare and to the best of our knowledge only one other research effort (apart form our own preliminary research investigations using direct seller-buyer features and machine learning approaches [Trattner et al., 2014]) exists so far.

The study of Guo et al. [2011] was performed to investigate the predictive power of social features such as direct and indirect interactions between sellers and buyers on the Chinese website Taobao (one of the world's largest electronic marketplaces) to recommend sellers to buyers. Among other things, the authors find that direct seller-buyer interactions and product meta-data information are the best features to tackle the task. Although, their work is similar to our own one, many significant differences can be found.

First, contrary to our study, the work of Guo et al. [2011] relies on social network data that has been directly induced from interactions between users in Tabao. Compared to this, our study is based on features and interactions that were induced from an external social networking platform that is independent from the marketplace itself. Second, we study a much richer set of features induced from social information such as user check-ins, user interests, group joins, etc. Information that has to the best of our knowledge not been leveraged yet for this kind of task. Third, we use our features in the context of a user-based collaborative filtering method, a well-established and robust recommendation approach often used in e-commerce websites, while the study of Guo et al. [2011] uses a machine learning approach called RankSVM to generate a list of preferred sellers. Finally, we show the extent to which the induced features and similarity metrics are useful over time, a concept that to the best of our knowledge has been neglected yet in all of the related works.

3.2.4 Datasets

In our study we rely on two datasets obtained from the virtual world Second Life. The main reasons for choosing Second Life over real-world sources are manifold, but mainly due to the fact that currently there are no other datasets available that comprise marketplace and social data of users at the same time.

Marketplace Dataset

Second Life provides an online trading platform called Second Life Marketplace where Second Life users are able to trade with virtual goods. Similarly to online shopping platforms such as eBay a user can be a seller, a buyer, or both. To collect this kind of information we gathered all store sites of the Second Life Marketplace with a web crawler. This crawler detected 131,087 stores/sellers, whereof 36,330 had at least one product in supply and 17,914 sold at least one product (for our study we only relied on those). Overall 1,725,449 products in 22 different categories (e.g., *Avatar Accessories* or *Vehicles*), with different prices and user ratings were found, from which 120,762 were purchased at least once. The total number of noticed purchases was 268,852 with 77,645 different buyers. Due to the fact that a seller can also be a buyer and a buyer can also be a seller, 8,259 users acted as both seller and buyer. The total number of involved users was 87,300. This obtained data stretches from July 2005 to February 2013. A basic overview of the marketplace dataset is provided in Table 3.1. Linking all sellers with their buyers based on the product reviews was our basic idea for the marketplace network for the experiments in this paper. Figure 3.1 shows the purchase distribution for the marketplace users, the transacted purchases over time, and the distribution of overall users with the fraction of new ones in a period of more than seven years. It exhibits that the Second Life Marketplace became more popular over time since the absolute number of purchases ascends correspondingly.

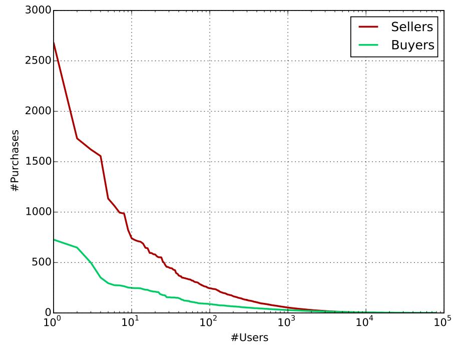
Online Social Network Dataset

Similarly to the real world, users in the virtual world of Second Life are able to establish social links through an online social networking platform called My Second Life. It was introduced by Linden Lab in 2007 and can be compared with other online social networks such as Facebook or Google+. This platform gives Second Life users the opportunity to present personal information on their user profiles or to interact with other users on the so-called Feed, which can be compared with the Timeline in Facebook. A considerable difference to Facebook exists concerning friendship relations. Such a relation type does not exist in My Second Life [Steurer and Trattner, 2013a].

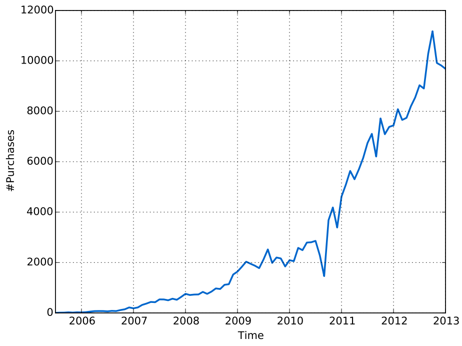
At the end of March 2013 we crawled the Second Life profiles of users who had not changed their profiles to private, based on the crawling

Table 3.1: **Marketplace Dataset Statistics.** This table shows the basic statistics of the marketplace dataset.

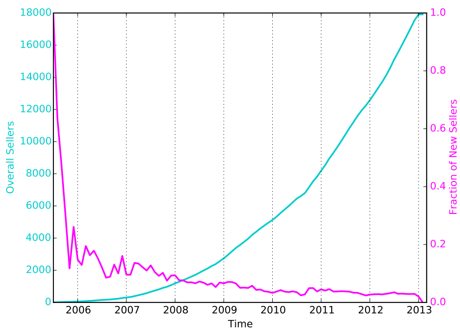
#Users	87,300
#Trading Interactions	268,852
#Trading Relations	219,889
#Sellers	17,914
#Buyers	77,645
#Sellers+Buyers	8,259
#Product Categories	22
#Products	120,762
Average #Products per Seller	≈7
Average #Purchases per Seller	≈15
Average #Purchases per Buyer	≈3



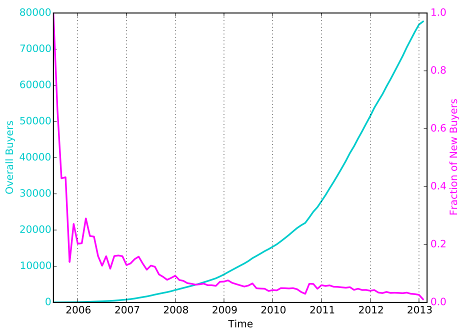
(a) Purchase distribution of the marketplace users.



(b) Purchase distribution over time.



(c) Overall sellers and the fraction of new sellers over time.



(d) Overall buyers and the fraction of new buyers over time.

Figure 3.1: **Distributions of Purchases, Sellers, and Buyers.** These plots show the distributions of purchases, sellers, and buyers of the marketplace dataset.

methodology as described in our previous work [Trattner and Steurer, 2015]. For each user we obtained the stated interests, the joined groups, the feed interactions with others (text messages, pictures, comments, likes), and the preferred in-world locations—so-called favored regions. Also in-world check-ins can be shared, which is a similar concept to Foursquare check-ins in Facebook. The basic statistics of this dataset are available in Table 3.2.

3.2.5 Experimental Setup

In this section we provide a detailed description of our experimental setup. First, we describe the recommender approaches used to tackle the task of recommending sellers to buyers. After that, we introduce the similarity features we have chosen in the two provided datasets, that form the basis of our recommendation approach in order to tackle the task of recommending sellers to buyers. Finally, we describe the evaluation methodology and the metrics used in our study.

Recommendation Approaches

Baseline. As baseline we chose a simple most popular recommender approach that recommends the most popular sellers to a potential buyer. Popularity was computed in terms of the number of purchase transactions the user performed.

User-Based Collaborative Filtering. The main approach we adopted in order to tackle the task of recommending sellers to buyers is a user-based collaborative filtering approach [Liben-Nowell and Kleinberg, 2007]. The basic idea of this approach is that buyers who are similar to each

Table 3.2: **Online Social Network Dataset Statistics.** This table shows the basic statistics of the online social network dataset.

#Users	152,509
#Postings (Text Messages/Pictures)	226,668
#Comments	348,106
#Likes	1,494,044
#Group Joins	1,869,281
#Stated Interests	227,596
#Check-in Postings	466,930
#Unique Check-in Regions	13,251
#Users with Check-ins	36,430
#Stated Favored Regions	337,732
#Unique Favored Regions	22,742
#Users who stated Favored Regions	76,093

other will behave in a similar manner in the marketplace [Schafer et al., 2007]. Out of the different collaborative filtering approaches, we used the non-probabilistic user-based k-nearest neighbor algorithm, where for each user we first find the k-nearest similar users and create a ranked list of their sellers. Afterwards, we recommend only the top-N sellers of the list that are new to the target user (i.e., the user is not a customer of these sellers).

In particular, we calculated the similarity values between the user pairs $sim(u, v)$ based on the user similarity features proposed in Section 3.2.5 (e.g., constructing the neighborhood). We defined the k-nearest neighbors of a buyer b as $neighbors(b)$ and the coefficient $S_{s,b}$ is 1, if b is a customer of seller s, and 0 otherwise. Based on these values, we ranked each seller s of the k most similar buyers to b using the following formula [Schafer et al., 2007]:

$$pred(b, s) = \sum_{n \in neighbors(b)} sim(b, n) \cdot S_{s,n} \tag{3.1}$$

In our experiments we applied various numbers for the parameters k and N. In this paper we only present the results with the best performance of our collaborative filtering approach that was obtained when setting $k = 100$ and $N = 5$ (see Section 3.2.6).

Similarity Features

In this section we describe in detail the features we induce from our two datasets which form the basis for our user-based collaborative filtering approach as introduced in the previous section. Utilizing different features from different data sources in our collaborative filtering method not only allows us to compare the predictive power of each feature but also helps us to understand what type of data source (in our setting online social network vs. marketplace data) is the most valuable one. As shown in our previous work [Steurer and Trattner, 2013b], similarities between users can be derived in two different ways. Either we calculate similarities between users on the content (e.g., user interests, products purchased, or groups they joined, denoted further as *homophilic features*), or on the network structure of the respective network, denoted as *network features*. In the following, we describe the types of similarity features we inducted from the Second Life Marketplace dataset and from the Second Life online social network.

Network Features. As features for the structure of a network we used the following measures, where $N(u)$ are the neighbors of a user u in the network. We denoted incoming neighbors as $N^{-}(u)$ and outgoing neighbors as $N^{+}(u)$:

Adamic Adar [Adamic and Adar, 2003; Cheng et al., 2011]:

$$sim(u,v) = \sum_{z \in N^{-}(u) \cap N^{-}(v)} \frac{1}{log(|N^{-}(z)|)} \tag{3.2}$$

Jaccard's Coefficient [Cranshaw et al., 2010; Steurer and Trattner, 2013a; Steurer et al., 2013]:

$$sim(u,v) = \frac{|N(u) \ \cap \ N(v)|}{|N(u) \ \cup \ N(v)|} \tag{3.3}$$

Preferential Attachment Score [Barabasi and Albert, 1999; Cheng et al., 2011]:

$$sim(u,v) = |N^{-}(u)| \ \cdot \ |N^{+}(v)| \ \text{ (or vice versa)} \tag{3.4}$$

Interactions [Steurer and Trattner, 2013a]:

$$sim(u,v) = |interactions(u,v)| \tag{3.5}$$

Reciprocity [Cheng et al., 2011]:

$$sim(u,v) = \begin{cases} 1 & \text{if link in both directions} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

The *Jaccard's Coefficient* feature of the *network features* was split into incoming and outgoing features. This means that, either only the incoming neighbors or only the outgoing neighbors of the users in the network were considered.

Homophilic Features. We constructed the following content-based similarity features, where $V(u)$ is a vector of items of a user u.

Jaccard's Coefficient:

$$sim(u, v) = \frac{|V(u) \ \cap \ V(v)|}{|V(u) \ \cup \ V(v)|} \tag{3.7}$$

This feature was applied in the case of the Second Life online social network to the user's interests, joined groups, check-ins, and favored regions.

Cosine Similarity [Steurer and Trattner, 2013a]:

$$sim(u, v) = \frac{V(u) \ \cdot \ V(v)}{\|V(u)\| \ \|V(v)\|} \tag{3.8}$$

This measure was used in the case of the Second Life Marketplace to the user's product categories, product prices, and product ratings.

Evaluation

The evaluation protocol we followed in this paper is one usually used in order to evaluate a recommender system offline in a time-based manner [Campos et al., 2014]. First, we considered in our evaluation only users who were present in both datasets (marketplace and online social network) in order to have a fair comparison of the two data sources. Second, for the sellers to buyers recommendation task we considered only those sellers as relevant who have not been observed by the buyer before (i.e., we only recommend sellers to buyers with no trading transactions in the past). Figure 3.2 presents the mean fractions of sellers who are new to the buyer or system over time. As shown, a huge fraction of sellers (over 60%) are always new to the buyer showing the potential of a seller to buyer recommender systems.

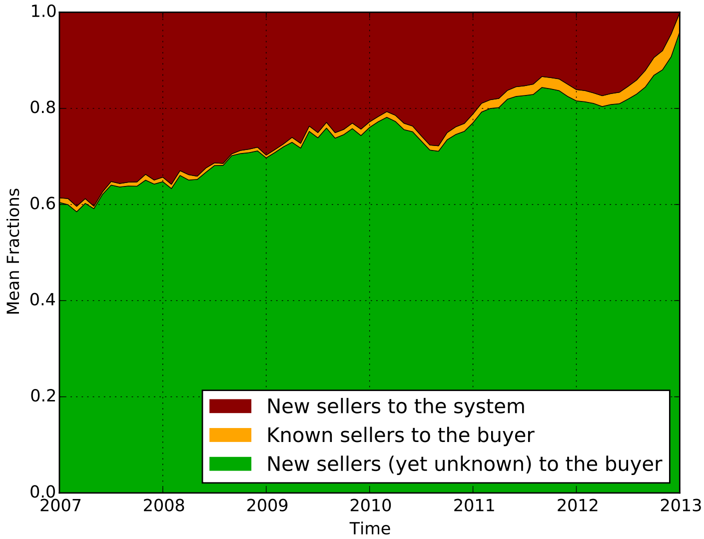


Figure 3.2: **Mean Fraction of Sellers.** This plot shows the mean fractions of sellers who are new to the buyer or system over time. As shown, over 60% of the sellers are new to the buyers and only a few sellers are known to them. This trend is increasing in time, showing the potential of a recommender system that recommends sellers to buyers in the Second Life Marketplace.

We split the Second Life Marketplace dataset in training and test samples according to the timeline. Consequently, we did the same with the Second Life online social network. The methodology we follow here is to train our recommender on all historical data available at some point in time t and to use the next forthcoming n months in time for testing. In particular, we generated recommendations every month over the time line (using all historical purchase events for training) beginning in 2007 until 2013 and used the purchase events of the next six forthcoming months for testing.

Figure 3.3 shows the sizes of the training and test sets with respect to the number of purchases and the number of buyers for whom a recommendation can be computed over time. Since the available data is very sparse at the beginning of our timeline, we consider only the results between 2009 and 2013.

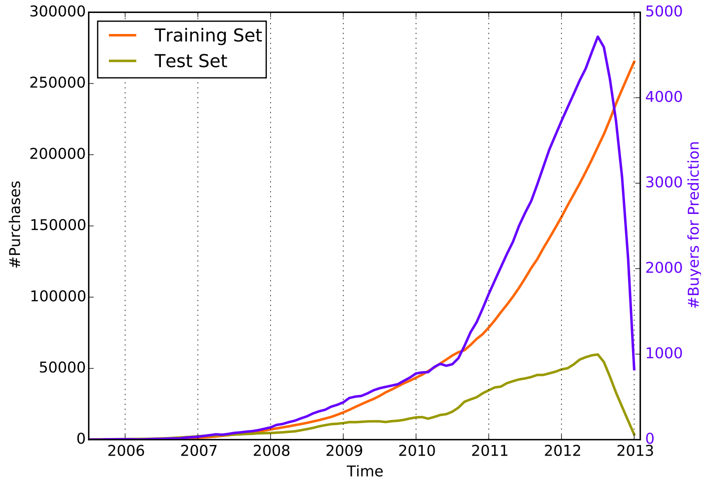


Figure 3.3: **Training and Test Set Sizes.** This figure depicts the sizes of the training and test sets and the number of buyers for whom a recommendation can be computed over time. As shown, until 2009 training and test sets are relatively small.

In order to determine the predictive power of our recommendation approach two evaluation metrics typically used in recommender systems were employed. In particular, we used the F1 score@5 and the user coverage to show the extent to which the corresponding similarity features and datasets perform [Herlocker et al., 2004].

3.2.6 Results

In this section we present the results of our experiments. First, we show how the datasets and the corresponding induced similarity features from the Second Life online social network and the Second Life Marketplace perform in the context of our user-based collaborative filtering approach for the task of recommending sellers to buyers—here we are interested in the social data source and the corresponding social information (RQ 1). After that, we show how well these features perform over time (RQ 2).

RQ 1: Recommending Sellers to Buyers

Figure 3.4 shows the mean values of the F1 score (left y-axis) for each used feature (x-axis) with the respective user coverage (right y-axis) from 2009 to 2013. As shown, *homophilic features* such as *Groups Jaccard* or *Interests Jaccard* as found in the social network are very valuable similarity features in a user-based collaborative filtering setting to recommend sellers to buyers efficiently compared to the most popular baseline. They are even to the same extent useful as historical features, such as *CosSim Product Categories* induced directly form the Second Life Marketplace. Interestingly, when comparing location-based social features to a most popular baseline, *Favored Regions Jaccard* just show little improvement, while *Check-ins Jaccard* could not improve the results. This is in line with previous observations that people in virtual worlds are not bound to certain places due to the possibility to teleport to places [Balby Marinho et al., 2015].

Note that the user coverage for the most popular baseline is under 100%. This can happen, since in our recommender task we only consider sellers which are not yet known to the buyer (see Figure 3.2).

RQ 2: Feature Performance over Time

Figure 3.5a shows the *network features* of the online social network over time. Although the performance of the features varies over time, it indicates that most of our *network features* of the social dataset are above the baseline at each point in time. As Figure 3.5b shows, the joined groups and stated interests are powerful information regarding sellers to buyers recommendations. One potential explanation for the improving performance of the *Groups Jaccard* feature compared to the most popular baseline over time could be the increasing amount of data available (see Figure 3.3).

As expected, the values of all features of the marketplace network are above the baseline most of the time as Figures 3.5c and 3.5d reveal. The *Preferential Attachment Score*+− feature of the *network features* and the *CosSim Product Categories* feature of the *homophilic features* are

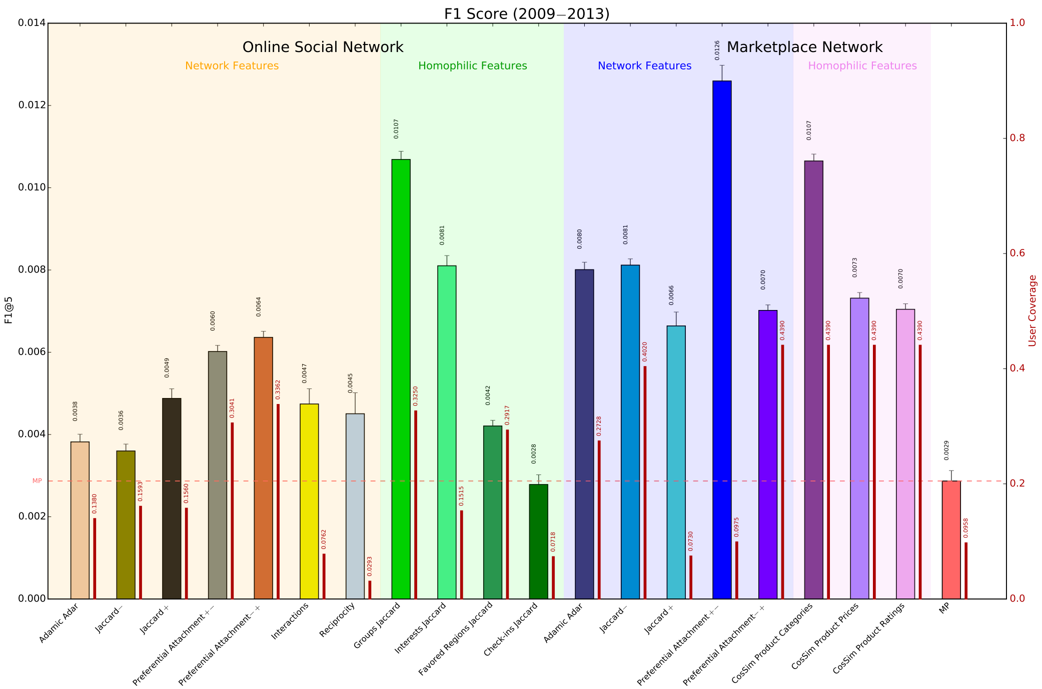
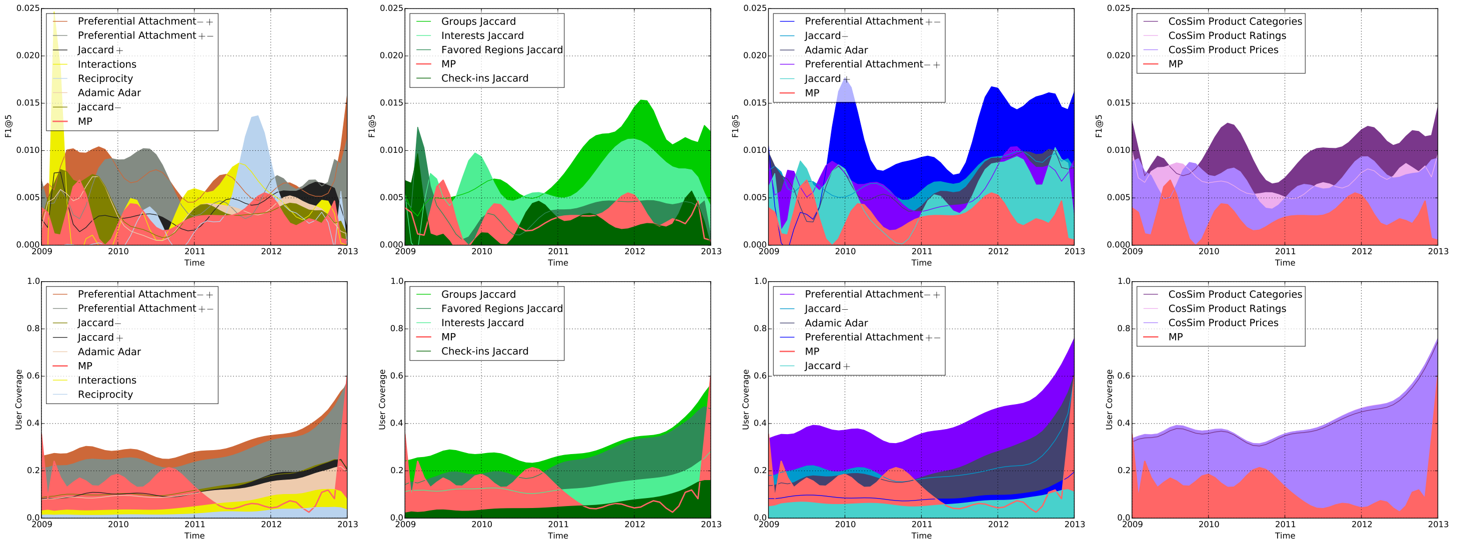


Figure 3.4: **Individual Feature Performances.** This plot shows the mean values of the F1 score of the similarity features and their respective user coverage over four years. As shown, *homophilic features* such as *Groups Jaccard* or *Interests Jaccard* as found in the social network are very valuable similarity features in a user-based collaborative filtering approach to recommend sellers to buyers efficiently compared to the most popular baseline (dashed line for comparison). They are even to the same extent useful as historical features such as the *CosSim Product Categories* induced directly from the Second Life Marketplace.



(a) *Network features* of the online social network.
(b) *Homophilic features* of the online social network.
(c) *Network features* of the marketplace network.
(d) *Homophilic features* of the marketplace network.

Figure 3.5: **Feature Performances over Time.** These plots depict the F1 score and user coverage for the induced similarity features in the different data sources (online social network and marketplace network) over time. As shown, the *network features* of the online social network and the trading network oscillate over time while *homophilic features* behave more stable. Furthermore, some trends over time can be observed. Features such as *Groups Jaccard*, *Preferential Attachment+−* of the online social network and *CosSim Product Categories* of the marketplace perform better over the years.

the most suitable features for sellers predictions. Except for the—for us unaccountable—peak at year 2010, both features also become more suitable for our recommendation task from time to time.

As shown in Figure 3.5, the user coverage of the most popular approach slightly decreases over time. The reason for this behavior is the strong increase of buyers in the system in 2011 (see Figures 3.1d and 3.3).

3.2.7 Conclusions

In this paper we extended our understanding of the signals available in social networking sites for the task of recommending sellers to buyers in dynamic online marketplaces. We approached this by conducting several offline experiments over time by employing a user-based k-nearest neighbor collaborative filtering method using several user similarity metrics that have been derived from social information such as likes, comments, joined groups, checked-in places, or stated interests. As our experiments reveal, most types of the social information we used are useful for the task of recommending new sellers to buyers in online marketplaces. Furthermore, we find that the methods vary significantly over time raising the question, if better time-dependent alternatives can be found that better adapt to the statistical properties of our dataset.

Limitations and Future Work. One of the limitations of our study is that we conducted our experiments only on one dataset. Applying our methods to other types of datasets would be an interesting extension of our work. Another limitation are the features for the predictions task, for which we believe better time-dependent alternatives could be found [Tylenda et al., 2009]. Finally, it would be interesting to apply machine learning to this kind of recommendation task (e.g., in the form of a learning to rank method that combines features) [Macedo et al., 2015] and to study the extent to which direct (as proposed in our previous work [Trattner et al., 2014]) vs. indirect features compare with each other.

3.3 Predicting Trading Interactions in an Online Marketplace Through Location-Based and Online Social Networks

This article also addresses the first research question and focusses on investigating the link prediction problem for trading interactions simultaneously in multiplex networks. Specifically, my co-authors and I investigate the extent to which trading interactions between sellers and buyers within an online marketplace platform can be predicted based on three different but overlapping networks, an online social network, a location-based social network and a trading network. Therefore, we evaluate a vast number of topological, especially path-based, as well as homophilic features and feature sets, and combinations of the networks, in order to determine the respective predictability. We consider both unsupervised as well as supervised methods.

The results of this article suggest that features of the online social network and the location-based network are to some extend useful and achieve suitable results, whereas the best trading interaction prediction results are achieved by features from the trading network. Due to the strong predictive power of the trading network information, the addition of further information of other network sources is not necessarily required. But, in specific settings, for example, for cold-start predictions or in case that trading information is not available, online and location-based social network information on their own or in combination could lead to convenient results at an acceptable scale. Further, we find that topological features are better suited than homophilic features for the recommendation of trading interactions, meaning that the information exploited from the network structure is more useful than other user related attributes represented through homophilic features.

3.3.1 Abstract

Link prediction is a prominent research direction for, for example, inferring upcoming interactions to be used in recommender systems. Although this problem of predicting links between users has been extensively studied in the past, research investigating this issue simultaneously in multiplex networks is rather rare so far. This is the focus of this paper. We investigate the extent to which trading interactions between sellers and buyers within an online marketplace platform can be predicted based on three different but overlapping networks—an online social network, a location-based social network, and a trading network. In particular, we conducted the study in the context of the virtual world Second Life. For that, we crawled according data of the online social network, user information of the location-based social network obtained by specialized bots, and we extracted purchases of the trading network. Overall, we generated and used 57 topological and homophilic features in different constellations to predict trading interactions between user pairs. We focused on both unsupervised as well as supervised learning methods. For supervised learning, we achieved accuracy values up to 92.5%, for unsupervised learning we obtained nDCG values up to over 97% and MAP values up to 75%.

3.3.2 Introduction

Social networks capture useful information about the relations between their users and their social characteristics [Coleman, 1988]. Since also the evolution of that structure is of particular interest, a large part of recent research activity in social networks is related to the link prediction problem: Here the goal is to estimate, whether two users u and v will interact with each other in the future or not [Liben-Nowell and Kleinberg, 2007]. Most of the work in this area being applied, for example, for friend recommendation [Barbieri et al., 2014], or community recommendation [Backstrom and Leskovec, 2011]. Recent approaches also include multiple relations that cover social networks from different perspectives, for example, considering different relations between the set of actors, in order to enable context-aware social personalization and recommendation systems [Eirinaki et al., 2018]. However, work that provides insights in which source of information is the most useful one, what types of features shall be used, and how well do both of these perform in unsupervised as well as supervised settings, are rare.

Objective. The problem addressed in this paper is a particular kind of link prediction problem—relevant both for virtual as well as physical social networks. Here, we want to show if and to what extent it is possible to predict who will buy from whom and who will sell to whom, or in other words, who will trade with whom in the future. The predictions are based on three sources of data: (i) an online social network, (ii) a location-based social network, and (iii) a trading network, including topological and homophilic features of these three different networks.

Here, we aim at extending the existing analyses in two directions: we focus on a combined feature-based analysis of the different networks, in contrast to, for example, Guo et al. [2011] or Zhang and Pennacchiotti [2013], we specifically focus on the impact of different features since these provide actionable insights that can be used for decision making later. We aim at predicting trading interactions between users from four perspectives: an online social, a location-based social, a trading network, and different combinations of them in order to see, if this increases our prediction results. We also consider both unsupervised as well as supervised methods, applying different constellations of features generated by the set of networks. This is useful for estimating performance indicators towards real application [Bischoff, 2012], and it also provides additional support with respect to the importance of features (and their combinations).

As a data source for our experiments we rely on Second Life[1]: it aims to provide users a platform similar to the real world but virtually to interact with each other via a social network called My Second Life. This also allows to create businesses over the Second Life marketplace that is similar to what we refer to as eBay[2] in the real world, as shown in Szell et al. [2012], Lehdonvirta [2009], and Guo et al. [2011].

[1] https://www.secondlife.com
[2] https://ebay.com

Research Questions. To drive our research we have defined the following three high-level research questions, which we will investigate and discuss in the following sections:

RQ 1. First, we focus on individual features: to what extent can trading interactions be predicted based on features from a set of networks (social, location-based, trading) individually, considering local (proximity), path-based, and content-based features on the (overall) prediction accuracy?

RQ 2. Second, we focus on the different feature types, as well as their collective interplay: does the combination of different feature types across multiple networks (social, location-based, trading) increase the results of predicting trading interactions?

RQ 3. Third, we focus on different classification approaches and address the question about on different prediction approaches—unsupervised as well as supervised learning methods: what is their impact with respect to different constellations of the available features?

Outline. In the following sections we will review appropriate background literature, introduce the datasets and methodology chosen to address our research questions, and present and discuss the results of our study. Finally, we draw conclusions, discuss the limitations of our study and propose future research directions.

3.3.3 Background

Link prediction is a prominent method for link analysis in social networks. It aims at predicting *new* and *recurring* links between the involved actors [Liben-Nowell and Kleinberg, 2003; Getoor and Diehl, 2005; Liben-Nowell and Kleinberg, 2007; Al Hasan and Zaki, 2011; Zhang and Philip, 2014; Wang et al., 2015; Martínez et al., 2016]. However, little work has been done in the context of predicting interactions between sellers and buyers. Below, we first discuss related approaches for link prediction outlining both unsupervised as well as supervised approaches, before we describe the relation to feature engineering for link prediction: this includes network proximity as well as path-based and content-based measures. According to these dimensions, we summarize the different foci of link prediction approaches in Table 3.3. This section concludes with a final subsection, where we outline current gaps in the literature, and summarize differences to previous research. Furthermore, we provide and discuss a detailed list of contributions of this work.

Table 3.3: **Overview of Related Work.** This table gives an overview on (general) unsupervised and supervised methods for link prediction, as well was the utilized sets of features, according to the categorization in feature engineering discussed below.

Authors	Learning Type	Method	Topo-logical	Path Based	Homo-philic
Backstrom and Leskovec [2011]	Supervised	DT, LR, SRW	×	×	
Barbieri et al. [2014]	Supervised	JSVD, WTFW			×
Cheng et al. [2011]	Supervised	DT, LR	×	×	
Cranshaw et al. [2010]	Supervised	AB, RF, SVM	×		
Eberhard and Trattner [2016]	Unsupervised	CF	×	×	
Fire et al. [2011]	Supervised	AB, ANN, B, C4.5, kNN NB, SVM, RF, RoF	×		
Fire et al. [2013]	Supervised	B, J48, RF	×	×	
Hasan et al. [2006]	Supervised	B, DT, kNN, MP, NB, RBFN, SVM	×		
Jones et al. [2013]	Supervised	RF, SVM	×		×
Kibanov et al. [2015]	Unsupervised	CB	×		×
Leskovec et al. [2010]	Supervised	LR	×		
Liben-Nowell and Kleinberg [2007]	Unsupervised	CF	×	×	
Lichtenwalter et al. [2010]	Un-/Supervised	CF/B, J48, NB	×	×	
Lichtenwalter and Chawla [2011]	Un-/Supervised	CF/any in WEKA[†]	×	×	
Lichtenwalter and Chawla [2012]	Supervised	B	×		
Lu et al. [2010b]	Un-/Supervised	Hybrid	×	×	
Lü and Zhou [2010]	Unsupervised	CF	×		
Menon and Elkan [2011]	Supervised	MF	×	×	
Murata and Moriyasu [2007]	Unsupervised	CF	×		
Rowe et al. [2012]	Supervised	LR	×		×
Scellato et al. [2011]	Supervised	J48, MT, NB, RF	×		
Scholz et al. [2013a]	Unsupervised	URW	×		×
Scholz et al. [2012]	Unsupervised	CF	×	×	
Scholz et al. [2013b]	Unsupervised	CF, URW	×	×	
Steurer and Trattner [2013a]	Un-/Supervised	CF/J48, LR, SVM	×		×
Steurer and Trattner [2013c]	Supervised	LR, RF, SVM	×		×
Thiele et al. [2018]	Supervised	S/MC	×		
Zhang and Pennacchiotti [2013]	Supervised	LR, NB, SVM			
Zhang et al. [2013]	Supervised	SVM	×	×	×
Zhuang et al. [2012b]	Semi-Supervised	PLP-FGM, SVM, TPFG	×		

[†] Any supervised method available in WEKA, AB = AdaBoost, ANN = Artificial Neural Networks, B = Bagging, CB = Content-based Filtering, CF = Collaborative Filtering, DT = Decision Tree, J48 = Pruned or Unpruned of C4.5 Decision Tree, JSVD = Joint Singular Value Decomposition, kNN = k-Nearest Neighbors, LR = Logistic Regression, NB = Naïve Bayes, MF = Matrix Factorization, MP = Multilayer Perceptron, MT = Model Trees, PLP-FGM = Partially-Labeled Pairwise Factor Graph Model, RBFN = Radial Basis Function Network, RF = Random Forest, RoF = Rotation Forest, SRW = Supervised Random Walk, SVM = Support-Vector Machine, S/MC = Siena/Markov Chain, TPFG = Time-Constrained Probabilistic Factor Graph Model, URW = Unsupervised Random Walk, WTFW = Who to Follow and Why

Link Prediction Methods

The prediction of (new) links between nodes in a social network is an interesting and challenging task. A first comprehensive fundamental analysis was done by Liben-Nowell and Kleinberg [2003]. In particular, Liben-Nowell and Kleinberg [2003, 2007] defined the link prediction problem as the search to carefully predict edges that will be added to a given snapshot of a social network during a given interval, using network proximity measures. Such link predictions could be used for suggesting promising interactions between two individuals in such a social network [Zhang et al., 2013; Barbieri et al., 2014]. This work is concerned with the prediction of trading interactions using several user information sources similar to Guo et al. [2011].

In the link prediction literature, typically two different types of learning approaches are distinguished: unsupervised and supervised approaches. In the following two paragraphs, we review literature in these two strands of research.

Unsupervised Approaches. Extending the fundamental work of Liben-Nowell and Kleinberg [2003, 2007] using network proximity and path-based measures, Murata and Moriyasu [2007] investigated weighted variants of the network proximity measures Adamic Adar, common neighbors, and preferential attachment; essentially these methods obtain a ranking utilizing collaborative filtering techniques for obtaining similar users for a given user. Furthermore, Lü and Zhou [2010] presented an approach to analyze the role of weak ties in social networks, while Zhuang et al. [2012b] used active learning for inferring social ties.

Most of these works analyzed the predictability of new links in online social networks like co-authorship in DBLP or arXiv.org. The prediction of new links in real-world social contacts has been largely neglected. Zhuang et al. [2012a] present prediction techniques using location-based proximity as a—weak—proxy for face-to-face encounters and online social networks. In contrast, Scholz et al. [2012] conducted a first analysis concerning the predictability of new links in real face-to-face contact networks. In Scholz et al. [2013a], a method for link prediction on multiplex networks, based on the idea of link prediction using the rooted PageRank algorithm [Liben-Nowell and Kleinberg, 2007] is described yielding the hybrid rooted PageRank algorithm. This algorithm enables a combined inference on the multiplex network for prediction. In a similar setting, Kibanov et al. [2015] apply content-based filtering.

Also, a structural view on link prediction is taken in Scholz et al. [2013b]. In addition, the integration of heterogeneous information for link prediction is investigated in Scholz et al. [2014]. Furthermore, Lichtenwalter et al. [2010] as well as Lichtenwalter and Chawla [2011] introduced a novel unsupervised method, more precisely a restricted variant of rooted PageRank, and a new supervised method [Lichtenwalter and Chawla, 2012] for link prediction. Here, we extend these approaches covering both supervised and unsupervised methods.

Supervised Approaches. In the literature supervised learning is commonly used to predict links between users in a network whenever label information is available, employing various machine learning methods. Hasan et al. [2006] considered a social network with interactions as edges representing the co-authoring of research articles. Each article included at least author information and publication year. For link prediction, they first split the set of publication years into two non-overlapping sub-ranges as training and test set. Their classification dataset consisted of author pairs that already existed in the training set, but did not publish any papers together in this period. To become a positive example for their experiment, those author pairs had to publish at least one paper in the test set period, otherwise they represented a negative example. Each positive example of author pairs established a link between them, which did not exist for the period of the training set. Consequently, they had a binary classification problem that was solved by supervised learning. Here, they mainly focused on topological (proximity) features.

Backstrom and Leskovec [2011] introduced a supervised method, based on supervised random walks for predicting new links, focusing on the network structure as well. Similarly, Menon and Elkan [2011] present a supervised approach using matrix factorization. Lu et al. [2010b] applied a supervised approach using multiple sources, focusing on feature engineering methods.

Compared to these approaches, we provide a much more feature rich setting integrating multiple relations and feature sets into our prediction approach. In addition, Scellato et al. [2011] present a supervised learning framework integrating place features on location-based social networks. In this paper, we extend the approaches mentioned above by not only focusing on network (proximity) features or integrating place features, but by taking a more comprehensive view: We focus on user, homophilic, and locational features, integrating them in order to assess their impact and efficacy. Thiele et al. [2018] present a longitudinal analysis of social network data for link prediction in the scope of the predictive impact of initial face-to-face contacts on the formation and evolution of developmental peer network relationships. For the predictive model, a longitudinal RSiena [Ripley et al., 2011] model is applied. In the context of this paper, we instead focus on the relations between the different networks, considering the combinations of features for predicting seller-buyer interactions.

Overall, several machine learning algorithms for the supervised link prediction in online social networks have been investigated in the past. For example, decision trees (C4.5 in the J48 implementation of WEKA [Hall et al., 2009]) were used in Cheng et al. [2011] and Fire et al. [2011], logistic regression in Cheng et al. [2011], Jones et al. [2013], Leskovec et al. [2010], Rowe et al. [2012], and Zhang and Pennacchiotti [2013], random forest in Fire et al. [2013] and Jones et al. [2013], and support-vector machines in Fire et al. [2011], Hasan et al. [2006], Jones et al. [2013], and Zhang and Pennacchiotti [2013], for a variety of tasks, as for example, reciprocal links, links of new users, or follower connections.

Features for Predicting Links Between Users in Networks

Within social networks, important information about users and their relations can be extracted in order to assess similarities between users. Topological and homophilic features are hypernyms for such user similarities in partly large-scale network data [Coleman, 1988; Steurer and Trattner, 2013a]. Then, these can be leveraged in link prediction approaches, relying on the (similar) social context of the users, as, for example, shown in Liben-Nowell and Kleinberg [2003] or Guo et al. [2011].

Topological Features. If the structure of a network is known, then network topological features can be applied for estimating the similarity between two users in the network, also in longitudinal analysis [Thiele et al., 2018].

For the analysis of co-authorship social networks, Liben-Nowell and Kleinberg [2007] used topological features for link prediction. They used measures such as common neighbors (number of neighbors that two users have in common), Jaccard's coefficient (number of common divided by number of total neighbors) as proposed by Salton and McGill [1983], Adamic Adar (regarding the node degree of the common neighbors) proposed by Adamic and Adar [2003] or preferential attachment (multiplication of numbers of neighbors of two users) proposed by Barabasi and Albert [1999], Newman [2001], and Barabasi et al. [2002]. More detailed topological feature measures were used by Steurer and Trattner [2013a]. They partly used a directed network for their experiments and thus distinguished between outgoing and incoming network topological features, for example, common neighbors, total neighbors, Jaccard's coefficient, and preferential attachment, were each split into an outgoing and an incoming feature. Furthermore, they applied the reciprocity of user communication, Adamic Adar, and the neighborhood overlap. Additionally, Fire et al. [2013] defined topological features such as transitive friends (number of outgoing neighbors of a user intersected by the number of incoming neighbors of another user), Katz measure (path oriented measure) proposed by Katz [1953], opposite direction friends (reciprocity between two users) for directed graphs or shortest paths.

Path-Based Features. As an extension of topological features that focus on the (local) neighborhood of a node, path-based features take into account richer connectivity information. The rooted PageRank [Liben-Nowell and Kleinberg, 2003] algorithm, as an adaption of the PageRank algorithm [Brin and Page, 1998] provides the stationary probability distribution sets of nodes regarding a specific starting node, providing a ranking for link prediction. Similarly, the Katz [Katz, 1953] measure also takes into account longer paths extending the neighborhood, weighted by a damping factor. Katz basically measures the strength of the connection between two nodes: The more paths two nodes are connected with and the shorter these paths are, the stronger the connection. For very small values of those, Katz is actually similar to a network proximity measure based on the nodes' neighborhood, because path lengths greater than three do not contribute very much [Liben-Nowell and Kleinberg, 2003].

Homophilic Features. Thelwall [2009] described homophily as the tendency for friend- or relationships to occur between individuals. Generally, homophily is the principle that an interaction between people rather occurs if they are similar than between dissimilar people. The target of homophily is to perceive and localize the behavioral, cultural, genetic, or material information that flows through networks.

Homophily structures the edges of a network of every type or relationship, which could be marriage, friendship, information transfer, work advice, or other types of relationships. For the personal environment common homophilic attributes are age, religion, education, occupation, and gender. However, homophilic attributes are very crucial for the user behavior, the information users receive, and the attitudes they form, as investigated by McPherson et al. [2001] in the context of online social networks. Steurer and Trattner [2013a] used attributes as groups, interests, user interactions, events, and regions for the computation of homophilic features for their experiments for predicting partnerships in social networks. For the different attributes they computed measures—also used in this paper—such as common items (number of items of an attribute two users have in common), total items (number of total items of an attribute of two users), Jaccard's coefficient (common items divided by the number of total items), or cosine similarity of the item vectors.

Feature Set Modeling. Overall, we can approach the link prediction task using different feature sets, dependent on their availability, both individually as well as in combination covering multiple networks. Although Fire et al. [2011] used only topological features in their link prediction experiments, they demonstrated that their models surprisingly achieved considerable results. Their goal was to predict hidden links in social network structures which they tried to achieve with machine learning methods applied on several social network datasets such as Academia, TheMarker, Flickr, Youtube, and Facebook. In order to obtain topological network features, the npreviousetwork structure has to be given. Otherwise, link prediction could be applied on homophilic features, which could also be a good measure for the similarity between the users in a network, for example as shown by Thelwall [2009]. They attained highly significant indications of homophily for measures such as ethnicity, age, religion, sexual orientation, country, or marital status for their experiments with a MySpace dataset. Furthermore, regarding the work of Cranshaw et al. [2010], they applied a hybrid approach, combining location-based data with online social network data. They used the location-sharing Facebook application called Locaccino and tried to predict the links in the online social network. Steurer and Trattner [2013a] also combined online social network data with location-based social network data in their partnership prediction experiments.

Differences to Previous Research and Contributions

In summary, the background literature discussed above clearly shows that the general link prediction problem is a well-studied area of research. Many studies have been performed to predict links in online social networks or other types of networks. However, surprisingly little work has been conducted employing several different networks at the same time, for instance concerning different types of networks, and in particular the combination of multiple networks, as well as feature types.

We started to investigate that topic using a location-based and an online social network in Steurer et al. [2013], Steurer and Trattner [2013c], and Kibanov et al. [2015] regarding interactions and their types. In addition, we tackled link prediction in the context of multiplex networks for predicting face-to-face interactions in Scholz et al. [2013a], and for attending talks in the context of academic conferences Scholz et al. [2014].

Furthermore, when reviewing the literature regarding the particular problem we study, namely predicting links (trading interactions) between sellers and buyers, we see that actually very little work can be found for that particular area. To the best of our knowledge there is only one study that is directly comparable to our work that has been performed by Guo et al.

[2011] in the past, apart from preliminary own work of authors of this paper [Eberhard and Trattner, 2016].

The work of Guo et al. [2011] is interesting as it is the first to study usefulness of social networks and 13 different features to predict seller-buyer interactions. The context of their work is the largest electronic marketplace in Chine named TAOBAO, with over 370 million registered users at the end of 2010. Among the features investigated, they employ centrality metrics, such as PageRank as well as homophilic metrics such as the number of common friends a seller and a buyer have in common to predict trading interactions. Also they used the prizes of the products as well as the ratings of the products as a proxy. The intention behind this is, that buyers typically buy from popular sellers (captured, for example, over centrality metrics such as PageRank) or keep also others types of types of relations, such as common friends that have been buying an item from the buyer before.

In this work we use similar features. However, compared to the work of Guo et al. [2011] we do not only rely on social network data, but also reveal whether there is also a signal present in the people's location-based network to predict trading interactions. The features engineered in our approach are based on the existing related work. This includes link prediction and recommender systems research as well as sociology as mentioned before, but also economy [DiMaggio and Louch, 1998] which suggests that the social embeddedness of the sellers in the buyers networks is inherently important for further purchase decisions. As such, we induce in total 57 different features capturing not only network effects between sellers and buyers, but we also consider homophilic features such as the number of groups or interests they have in common.

Furthermore, we make use of location-based network data, to understand whether features such as, for example, the number of times seller and buyers have been seen in the same location, bears a signal that can be exploited to predict seller-buyer links in the future. In addition to this, our experiments make use of different supervised and unsupervised learning approaches.

To the best of our knowledge, this is the first kind of study that shows the relation between and individual as well as combined impact of three different types of networks, features, features sets, and learning methods, to predict links between sellers and buyers.

In total, the contributions of this work can be summarized as follows:

1. The collection of a unique dataset of user and activity data in three different networks: an online social network, a location-based social network and a trading network.

2. The engineering of 57 topological and homophilic features to predict trading interactions between two users in these three different types of networks.

3. The statistical analysis of differences of the features to discriminate between seller and buyer interactions across different networks.

4. The presentation of a set of supervised and unsupervised learning experiments to show the meaningfulness of the inducted features individually to predict trading interactions between sellers and buyers.

5. The presentation of results revealing the value of predicting seller-buyer trading interactions based on feature sets (homophilic and topological) as well as different kinds of networks (trading, social, and location-based).

6. Finally, we show how the features correlate with each other and reveal their importance (measured via information gain) when considering all features at the same time in the model.

3.3.4 Datasets

In order to address the three research questions, it was necessary to have three different kinds of data available:

1. Data from an online social network, such as Facebook[3] or Google+[4], where users share personal information on their profiles and are able to communicate with others via the platform.

2. Data from a location-based social network, such as Foursquare[5], where geographical position information about the users is available.

3. Data from a trading network, such as eBay, where the community is able to trade with goods. The majority of these websites restrict the verbose crawling of their user profiles, but apart from this awareness, most of the users share their profiles only with their friends and prohibit the access by others.

As a consequence, we opted for the virtual world Second Life, which unites all the required kinds of data sources. On top of that, there is the advantage of a large user overlap across the three parts of the Second Life platform [Steurer and Trattner, 2013a,b,c; Steurer et al., 2013]. Although the users in Second Life do not interact with their real life names, but with the names of their avatars in a virtual world, La and Michiardi [2008] and Varvello et al. [2008] have shown that the avatars' behavior tends to be similar to the behavior of humans.

The basic principle of Second Life is that avatars explore the virtual world, meet other avatars, and communicate, play, or trade with them. Varvello and Voelker [2010] denoted the Second Life social network as small-world network and much more similar to a real-world network in comparison with popular online social networks. Crucial for this observation is the establishing of social relationships between users in Second Life, which requires an active interaction between the involved users. By contrast, relationships in online social networks often signify only the acceptance of a friendship request without existing interactions such as text messages between the users.

As source for our online social network we crawled data from My Second Life[6]. The location-based data were monitored in-world in Second Life and

[3] https://facebook.com
[4] https://plus.google.com
[5] https://foursquare.com
[6] https://my.secondlife.com

the Second Life Marketplace[7] was used as trading network in this paper. Table 3.4 provides an overview of the extracted entities of the respective data source that we further used for the feature engineering.

Online Social Network Data

As described in our previous work [Eberhard and Trattner, 2016], users in the virtual world of Second Life are able to establish social links through an online social networking platform called My Second Life. Similar to Facebook and its timeline, My Second Life gives Second Life users the opportunity to present personal information on their user profiles or to interact with other users on the so-called feed. Apart from such information about the Second Life avatar such as interests, the day of birth in Second Life, or the biography, users are able to join groups or to show their favorite in-world regions on their profiles. It is also possible to share text messages or pictures with others on the feed. Furthermore, these postings can be commented or loved. A "love" in Second Life is similar to a "like" in Facebook or a plus in Google+. A considerable difference to Facebook

[7] https://marketplace.secondlife.com

Table 3.4: **Extracted Entities.** This table shows the extracted entities from the three different data sources.

Online Social Network	**Location-Based Social Network**	**Trading Network**
Users	Users	Users
Interactions	Events	Product Categories
Postings	Event Categories	Product Prices
Text Messages	Event Regions	Product Ratings
Pictures		
Comments		
Loves		
Groups		
Interests		
Check-ins		
Favored Regions		

exists concerning friendship relations. Such a relation type does not exist in My Second life [Steurer and Trattner, 2013a].

Based on the crawling methodology described in our previous work [Trattner and Steurer, 2015], at the end of March 2013 we crawled the Second Life profiles of users with public user profiles. We extracted a list of user names from the location-based dataset (see Section 3.3.4) and iteratively extended it by further users who interacted on the feed with the users from the list. For each user, we obtained their interests, the joined groups, and the feed interactions with others. Two different sources of Second Life regions were also part of the collected information for each user. In Second Life it is possible to record in-world snapshots of regions in terms of pictures and share them on the feed to show others where users have actually been at a particular time. We collected these so-called check-ins for each user and thus count as personal user information. Besides the interests, groups, biography etc., the profiles in Second Life provide an area to state preferred in-world locations—the second source of locations and so-called favored regions.

We constructed the online social network on the basis of the feed interactions between the users, as an indicator for being acquainted. If the number of interactions was zero, no link was generated between them. Users with numbers of interactions greater than or equal to one were provided with an edge between them in the network. Eventually, this directed online social network was denoted as $G_O = \langle V_O, E_O \rangle$, where V_O was the set of users with interactions on their feeds. If a user $u \in V_O$ communicated with a user $v \in V_O$ by posting a text message on v's feed or commenting or loving a posting on v's feed, the edge between them was formally defined as $e = (u, v) \in E_O$.

First, this procedure reached a result of 169,035 users with 587,090 postings, 459,734 comments, and 1,631,568 loves, which gave a number of total interactions of 3,175,304. Due to the fact that this paper is about predicting trading interactions, self connections in the network have been removed, because seller and buyer are not the same person in a trading relation. In this way, the dataset of the online social network of Second Life slightly decreased. Now there were 152,509 users with 226,668 postings, 348,106 comments, and 1,494,044 loves, which gave a number of total interactions of 2,068,818. Probably, the number of loves remained nearly stable, because the loves for postings mostly apply to other users' postings and not one's own. Furthermore, the average of the number of interests defined by each user was ≈1.5 and the joined groups per user ≈12.4 on average. On average, for each user we obtained ≈3.1 check-ins on the feed and ≈2.2 favored regions. Table 3.5 gives an overview of the numbers of the online social dataset.

Table 3.5: **Statistics of the Online Social Network G_O.** This table shows basic statistics of the online social network dataset.

#Users	152,509
#Edges	270,567
Type	directed
Degree	3.55
#Connected Components	13,115
Largest Connected Component	77.69%
#Postings (Text Messages / Pictures)	226,668
#Comments	348,106
#Loves	1,494,044
#Overall Interactions	2,068,818
Average #Interactions per User	≈14
#Group Joins	1,869,281
#Unique Groups	204,769
#Users with Group Join(s)	114,205
#Stated Interests	227,596
#Unique Interests	62,170
#Users who stated Interest(s)	36,610
#Check-ins	466,930
#Unique Checked-in Regions	13,251
#Users with Check-ins	36,430
#Stated Favored Regions	337,732
#Unique Favored Regions	22,742
#Users who stated Favored Region(s)	76,093

Location-Based Social Network Data

We extracted the location-based dataset used in this paper from the in-world of Second Life by scripted robots collecting information about surrounding users. As described in our previous work [Trattner and Steurer, 2015], we sent the bots to locations of the Second Life event calendar from the Second Life website to presumably target regions with a higher user frequency than in other places in the huge world of Second Life. Starting from March 2012, over the period of one year, the collected user information formed the basis for the location-based social network. Overall, nearly 19 million data entries with 410,619 different users in 4,146 different locations were observed. To generate a network with an adequate density from this huge amount of data, we created a link between two users, if they had met each other more than only once. This is formally defined as $G_L = \langle V_L, E_L \rangle$, where V_L is the set of users and $e = (u, v) \in E_L$ the link between two users $u \in V_L$ and $v \in V_L$, if they were observed together in the same place at the same time on at least two different days. This rule reduced the number of edges in this network many times over to 1,414,389 and the number of nodes to 122,936. The total number of monitored events for all users was 1,966,206 with 81,671 unique events and 11 different event categories—an average of ≈16 events per user. There were 16,375,540 event regions entries registered with 3,972 unique regions, which means that on the average, each user was found ≈133 times by the bots. Table 3.6 provides an overview on the location-based dataset.

Trading Network Data

Besides the in-world of Second Life and My Second Life, there is an online trading platform called Second Life Marketplace where Second Life users are able to trade with virtual goods. The users can act as sellers, buyers, or both, similarly to common online shopping platforms such as eBay. Only if a purchase is done via the marketplace, the buyer can write a public review about the bought product or just rate the product from one to five stars. As a consequence, every stated review in the whole marketplace ensures the purchase of the product between the seller and the reviewer.

Linking all sellers with their buyers based on the product reviews was our basic idea for the trading network for the experiments in this paper.

Based on the crawling methodology described in our previous work [Eberhard and Trattner, 2016], we gathered all store sites of the Second Life Marketplace with a web crawler to collect the purchase information. This crawler detected 131,087 stores/sellers, whereof 36,330 had at least one product in supply and 17,914 sold at least one product. Overall 1,725,449 products in 22 different categories, for example avatar accessories or vehicles, were found, from which 120,762 were purchased at least once. The total number of noticed purchases was 268,852 with 77,645 different buyers. Due to the fact that a seller can also be a buyer and a buyer can also be a seller, 8,259 users acted as both seller and buyer. The total number of involved users was 87,300. An overview of the trading network dataset is provided in Table 3.7.

Table 3.6: **Statistics of the Location-Based Social Network G_L.** This table shows basic statistics of the location-based social network dataset.

#Users	122,936
#Edges	1,414,389
Type	undirected
Degree	23.01
#Connected Components	719
Largest Connected Component	98.01%
#Events Entries	1,966,206
#Unique Events	81,671
#Event Categories	11
Average #Events per User	≈16
#Event Regions Entries	16,375,540
#Unique Event Regions	3,972
Average #Entries per User	≈133

3.3.5 Methodology

In the previous sections we introduced the different sources of data used in our experiments. In this section we describe the methodology for the experiments used in this paper to answer the research questions.

Dataset Pre-Processing

To make the results for the different networks comparable, it was necessary to bring them on a common basis. We intersected the online social network, the location-based social network, and the trading network by picking out the common nodes of all networks. This means that we considered only those users who were active in all of the different networks, thus there was information about them in all of the network sources. Therefore, each user must have made at least one purchase as seller or buyer in the Second Life Marketplace, one interaction on My Second Life, and an in-world observation by the robots.

Table 3.7: **Statistics of the Trading Network G_T.** This table shows basic statistics of the trading network dataset.

#Users	87,300
#Edges	219,889
Type	directed
Degree	5.04
#Connected Components	933
Largest Connected Component	97.39%
#Sellers	17,914
#Buyers	77,645
#Sellers ∩ Buyers	8,259
#Product Categories	22
#Products	120,762
Average #Products per Seller	≈7
#Purchases	268,852
Average #Purchases per Seller	≈15
Average #Purchases per Buyer	≈3

We formally defined this combined network as $G_C = \langle V_C, E_C \rangle$, where V_C was the set of common users of the three networks, the online social network G_O, the location-based social network G_L, and the trading network G_T: $V_C = \{u \mid u \in V_O,\ u \in V_L,\ u \in V_T\}$. E_C was the union set of edges representing the relations between these users in either networks: $E_C = \{(u, v) \mid (u, v) \in E_O \text{ or } (u, v) \in E_L \text{ or } (u, v) \in E_T, \text{ and } u, v \in V_C\}$.

The numbers of this combined network, which we used for all the experiments in this paper, are shown in Table 3.8.

All basic computations for preparing the experiments were done with Python and NetworkX[8]. The experiments were completed in a way where the starting point is a random seller s. The prediction result should tell to what extent any random buyer b will buy from s based on appropriate features.

Feature Engineering

As mentioned in the background section (see Section 3.3.3), different characteristics can be extracted from networks and used for predicting links between users. In the following three subsection we describe in detail the features and feature sets which have been used and engineered to predict seller-buyer trading interactions. The first subsection describes how we

[8] https://networkx.github.io

Table 3.8: **Statistics of the Combined Network G_C.** This table shows basic statistics of the combination of all three networks.

#Users	10,420
#Online Social Network Edges	8,543
#Location-Based Social Network Edges	45,558
#Trading Network Edges	5,376
Total #Edges	59,477
#Sellers	2,086
#Buyers	9,655
# Sellers ∩ Buyers	1,321

inducted features from the online social network features, followed by a subsection detailing on how we induced features form the location-based social network. Finally, the trading network features are introduced. For each of the three data sources, we derive topological and homophilic features. While the former means inducing network-specific features the latter refers to features extracted from content.

Online Social Network: Topological Features. We defined the neighbors of a user u in this directed network with respect to the direction of the communication between them. A neighbor v that received messages from a user u is called outgoing neighbor and a neighbor v that sent messages to a user u is called incoming neighbor [Steurer and Trattner, 2013a]. We denoted the definition of outgoing neighbors of a user $u \in V_O$ as $\Theta^+(u) = \{v \mid (u, v) \in E_O\}$ and incoming neighbors as $\Theta^-(u) = \{v \mid (v, u) \in E_O\}$. Therefore, we could formally compute the whole set of neighbors of u as $\Theta(u) = \Theta^+(u) \cup \Theta^-(u)$.

#Common Outgoing Neighbors: We defined the number of neighbors that two users u and v have in common related to the outgoing communication of them as $O^+_{CN}(u, v) = |\Theta^+(u) \cap \Theta^+(v)|$. For example, a user $w \in O^+_{CN}(u, v)$ is a common outgoing neighbor of user u and v if both u and v sent one or more messages to w.

#Common Incoming Neighbors: This is the opposite of the common outgoing neighbors. The number of common incoming neighbors of two users u and v are the users who sent messages to both of them. We defined this feature as $O^-_{CN}(u, v) = |\Theta^-(u) \cap \Theta^-(v)|$. For example, a user $w \in O^-_{CN}(u, v)$ is a common incoming neighbor of user u and v if w sent one or more messages to u as well as v.

Outgoing Jaccard's Coefficient: The Jaccard's coefficient is the division of the number of common by the number of total neighbors of two users u and v and could be seen as a measure for exclusiveness of the relation between them [Cranshaw et al., 2010]. It was also split into an outgoing and an incoming feature. We denoted the outgoing Jaccard's coefficient as $O^+_{JC}(u, v) = \frac{|\Theta^+(u) \, \cap \, \Theta^+(v)|}{|\Theta^+(u) \, \cup \, \Theta^+(v)|}$.

Incoming Jaccard's Coefficient: This feature is the complement to the outgoing Jaccard's coefficient and we defined it as the number of common incoming neighbors divided by the number of total incoming neighbors: $O^{-}_{JC}(u,v) = \frac{|\Theta^{-}(u)\ \cap\ \Theta^{-}(v)|}{|\Theta^{-}(u)\ \cup\ \Theta^{-}(v)|}$.

Preferential Attachment +–: Here the preferential attachment score, first proposed by Barabasi and Albert [1999], is presented in a slightly different way, proposed by Cheng et al. [2011]. It is another popular measure to describe the correlation between the out-degree of a user u and the in-degree of a user v. We calculated the value for this feature as the product of the number of outgoing neighbors of u and the incoming neighbors of v, formally defined as $O^{+}_{PS}(u,v) = |\Theta^{+}(u)| \cdot |\Theta^{-}(v)|$.

Preferential Attachment –+: The difference to the preferential attachment $+-$ feature described above is, that the in- and out-degree of the involved users were swapped. So we denoted the preferential attachment in feature for two users u and v as $O^{-}_{PS}(u,v) = |\Theta^{-}(u)| \cdot |\Theta^{+}(v)|$.

Reciprocity of User Communication: The reciprocity of user communication in a directed network describes if a communication between two users u and v is bidirectional or in only one direction [Cheng et al., 2011]. We denoted this feature as
$O_R(u,v) = \begin{cases} 0 & \text{if } (u,v) \in E_O,\ (v,u) \notin E_O \\ 1 & \text{if } (u,v) \in E_O,\ (v,u) \in E_O \end{cases}$.

Adamic Adar: Regarding the relation between two users related to their neighbors, Adamic and Adar [2003] proposed a measure for the activity of the common neighbors of two users u and v in the network, because the definition regards the node degree of the common neighbors. For directed networks, Cheng et al. [2011] suggested a refinement of the Adamic Adar measure in which only the common incoming neighbors are considered:
$O_{AA}(u,v) = \sum_{z\ \in\ \Theta^{-}(u)\ \cap\ \Theta^{-}(v)} \frac{1}{log(|\Theta(z)^{-}|)}$.

Katz: Katz is a path-based attribute proposed by Katz [1953] that measures the strength of the connection between two nodes in a network. The more paths two nodes are connected with and the shorter these paths are, the stronger is the connection between the nodes. β expresses the emphasis of the path length l between two nodes u and v. The weight of shorter path lengths rises by decreasing β. Due to the high complexity calculating this measure for large networks, we introduced a cutoff $c = 3$ considering only paths with a maximum length of 3. With $|path^l_{u,v}|$ as the number of paths between u and v of length l, we formally defined the Katz measure as $O_{K\beta}(u, v) = \sum_{l=1}^{c} \beta^l \cdot |path^l_{u,v}|$.

Rooted PageRank: The rooted PageRank [Liben-Nowell and Kleinberg, 2007] is also a path-based measure and special kind of the personalized PageRank [Chakrabarti, 2007]. The rooted PageRank score between two nodes u and v is based on a random walk on the network starting at node u. With probability $1 - \alpha$ it iteratively moves to a random neighbor of the current node and with probability α it jumps back to u. We defined it as $O_{RPR\alpha}(u, v)$ = stationary probability of v under the following random walk [Pearson, 1905; Spitzer, 2013]: (i) with probability $1 - \alpha$ move to a random neighbor of the current node, and (ii) with probability α return to u.

Online Social Network: Homophilic Features. The groups a user u can join in this social network were defined as $\Delta(u)$ and the self-defined interests of u as $\Phi(u)$.

#Common Groups: This feature represents the number of groups two users u and v have in common: $G_C(u, v) = |\Delta(u) \cap \Delta(v)|$.

Jaccard's Coefficient of Groups: The Jaccard's coefficient as already mentioned can also be applied for homophilic measures such as groups, interests, regions, or events. In this case we denoted the Jaccard's coefficient for groups as $G_{JC}(u, v) = \frac{|\Delta(u) \cap \Delta(v)|}{|\Delta(u) \cup \Delta(v)|}$.

#Common Interests: The same types of features as defined for groups were determined for the interests users are able to declare on their social feed. The common interests feature shows the number of interests two users u and v have in common: $I_C(u, v) = |\Phi(u) \cap \Phi(v)|$.

Jaccard's Coefficient of Interests: For the user-defined interests we computed the Jaccard's coefficient for the proportion of common and total interests of two users u and v as $I_{JC}(u,v) = \frac{|\Phi(u) \cap \Phi(v)|}{|\Phi(u) \cup \Phi(v)|}$.

#Interactions: In the online social network of Second Life the users are able to share text messages with other users, or comment or love such messages. We defined the interactions from a user u to a user v as $\iota(u,v)$. So this feature shows the number of all interactions from u to v and we formally defined it as $OI(u,v) = |\iota(u,v)|$.

On the online social network feed of Second Life users are able to record in-world snapshots of regions in terms of pictures and share them to show their friends or followers where they have actually been at a particular time. Such regions a user u shared on the feed were denoted as $\Lambda(u)$.

#Common Check-ins: This feature is a measure for how many common regions two users u and v checked in and shared on their own feed and we formally specified it as $RR_C(u,v) = |\Lambda(u) \cap \Lambda(v)|$.

Jaccard's Coefficient of Check-ins: The value of the common check-ins divided by the value of the total check-ins of two users u and v is the Jaccard's coefficient measure again and we defined it as $RR_{JC}(u,v) = \frac{|\Lambda(u) \cap \Lambda(v)|}{|\Lambda(u) \cup \Lambda(v)|}$.

Overlap of Check-ins: The overlap of the sets of check-ins of two users u and v differs from the Jaccard's coefficient in terms of the division by the sum of u's and v's regions. We stated this feature as $RR_O(u,v) = \frac{|\Lambda(u) \cap \Lambda(v)|}{|\Lambda(u)| + |\Lambda(v)|}$.

Apart from interests, groups, or personal information, Second Life users are able to specify regions on their profiles. The purpose of such favored regions of users is to let others know about their preferred locations. The following features are based on these regions and the types of measures are again the same as from the check-ins. We defined the favored regions of a user u as $\Xi(u)$.

#Common Favored Regions: We defined the number of favored regions two users u and v have in common as $RF_C(u,v) = |\Xi(u) \cap \Xi(v)|$.

Jaccard's Coefficient of Favored Regions: We state the Jaccard's coefficient of favored regions of two users u and v as $RF_{JC}(u,v) = \frac{|\Xi(u) \cap \Xi(v)|}{|\Xi(u) \cup \Xi(v)|}$.

Overlap of Favored Regions: This feature represents the overlap between the common favored regions of two users u and v and the sum of the favored regions of u and the favored regions of v as $RF_O(u,v) = \frac{|\Xi(u) \cap \Xi(v)|}{|\Xi(u)| + |\Xi(v)|}$.

Location-Based Social Network: Topological Features. In the location-based social network we defined the neighbors of a user $u \in V_L$ as $\Gamma(u) = \{v \mid (u,v) \in E_L\}$. Similar to the topological online social network features described in Section 3.3.5, we subdivided features to measure the structural overlap of two users in the location-based social network as follows:

#Common Neighbors: This feature represents the number of neighbors two users u and v have in common. We denoted the common neighbors as $L_{CN}(u,v) = |\Gamma(u) \cap \Gamma(v)|$.

Jaccard's Coefficient: We stated the Jaccard's coefficient for two users u and v in the location-based social network as $L_{JC}(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|}$.

Adamic Adar: Slightly different from the Adamic Adar measure of the online social network described in Section 3.3.5, we formally defined the Adamic Adar for undirected networks as

$$L_{AA}(u,v) = \sum_{z \, \in \, \Gamma(u) \, \cap \, \Gamma(v)} \frac{1}{log(|\Gamma(z)|)} \; .$$

#Days Seen: We defined the set of days two users were in the same region at the same time as η. Therefore, the number of days two users u and v have met each other in-world was formally stated as $L_{DS}(u,v) = |\eta(u,v)|$.

Mean Distance: ω is the set of distances two users were apart in the same region at a certain point in time gathered by our bots. We stated this feature as $L_{MD}(u,v) = \frac{1}{|\omega(u,v)|} \sum_{d \, \in \, \omega(u,v)} d$.

Katz: As described for the online social network, Katz is a path-based measure that quantifies the strength of the connection between two nodes in a network based on lengths of the paths between them. For the location-based social network, again with a cutoff $c = 3$, we defined it as $L_{K\beta}(u,v) = \sum_{l=1}^{c} \beta^l \cdot |path_{u,v}^l|$.

Rooted PageRank: As mentioned above, the rooted PageRank is the stationary probability of node v based on a random walk that starts at node u. It iteratively moves to a random neighbor of the current node or jumps back to u. We formally denote it as $L_{RPR\alpha}(u,v)$, the stationary probability of v under the following random walk: (i) with probability $1 - \alpha$ move to a random neighbor of the current node, (ii) with probability α return to u.

Location-Based Social Network: Homophilic Features. As mentioned in Section 3.3.4 the implemented robots monitored users in-world at Second Life events. We stated the events a user u visited as $\Phi(u)$. The following features refer to such events and their locations:

#Common Events: We defined the number of common events which two users u and v visited as $E_C(u,v) = |\Pi(u) \cap \Pi(v)|$.

Jaccard's Coefficient of Events: We computed the Jaccard's coefficient measure of the events two users u and v visited as $E_{JC}(u,v) = \frac{|\Pi(u) \cap \Pi(v)|}{|\Pi(u) \cup \Pi(v)|}$.

Cosine Similarity of Event Categories: Another way to measure the similarity between two users u and v is to compute the cosine similarity of two vectors including some user specific attributes. In this case two vectors $\delta(u)$ and $\delta(v)$ with the length of the number of all categories of the Second Life events for each user pair (u,v) were defined. Every item i in such a vector represented the number of events the user visited of a specific category. We computed the cosine similarity of event categories between two users u and v as $E_{CCos}(u,v) = \frac{\delta(u) \cdot \delta(v)}{\|\delta(u)\| \ \|\delta(v)\|}$.

The information of the following features is based on the regions of the visited events of the users. The measures of the features are the same as from the check-ins and favored regions:

#Common Event Regions: We stated the number of regions of events two users u and v visited in common as $RE_C(u,v) = |\Upsilon(u) \cap \Upsilon(v)|$.

Jaccard's Coefficient of Event Regions: This feature measures the Jaccard's coefficient of the event regions of two users u and v: $RE_{JC}(u,v) = \frac{|\Upsilon(u)\ \cap\ \Upsilon(v)|}{|\Upsilon(u)\ \cup\ \Upsilon(v)|}$.

Overlap of Event Regions: We defined the overlap between the common event regions of two users u and v and the sum of the event regions of u and the event regions of v in this feature as $RE_O(u,v) = \frac{|\Upsilon(u)\ \cap\ \Upsilon(v)|}{|\Upsilon(u)|\ +\ |\Upsilon(v)|}$.

Trading Network: Topological Features. The topological features to measure the structural overlap of two users in the online social network described in Section 3.3.5 could also be applied on the trading network of the Second Life Marketplace. Since this network is directed, we split some of the features into outgoing and incoming features again.

We denoted the outgoing neighbors in the trading network of a user $u \in V_T$ as $\Psi^+(u) = \{v \mid (u,v) \in E_T\}$ and incoming neighbors as $\Psi^-(u) = \{v \mid (v,u) \in E_T\}$. The formal definition of the combined set of neighbors is then stated as $\Psi(u) = \Psi^+(u) \cup \Psi^-(u)$.

#Common Outgoing Neighbors: We defined the number of outgoing neighbors two users u and v have in common as $T^+_{CN}(u,v) = |\Psi^+(u) \cap \Psi^+(v)|$.

#Common Incoming Neighbors: The definition for the number of common incoming neighbors of two users u and v was stated as $T^-_{CN}(u,v) = |\Psi^-(u) \cap \Psi^-(v)|$.

Outgoing Jaccard's Coefficient: We denoted the definition of the outgoing Jaccard's coefficient of two users u and v of the trading network as $T^+_{JC}(u,v) = \frac{|\Psi^+(u)\ \cap\ \Psi^+(v)|}{|\Psi^+(u)\ \cup\ \Psi^+(v)|}$.

Incoming Jaccard's Coefficient: The incoming Jaccard's coefficient is the complement to the previous feature, given as $T^-_{JC}(u,v) = \frac{|\Psi^-(u)\ \cap\ \Psi^-(v)|}{|\Psi^-(u)\ \cup\ \Psi^-(v)|}$.

Preferential Attachment +–: As mentioned above the preferential attachment score is a measure for the correlation between the out-degree of a user u and the in-degree of a user v and we defined it as $T_{PS}^{+}(u,v) = |\Psi^{+}(u)| \cdot |\Psi^{-}(v)|$.

Preferential Attachment −+: The difference to the preferential attachment +− feature is the swapping of the users. We formally defined this feature as $T_{PS}^{-}(u,v) = |\Psi^{-}(u)| \cdot |\Psi^{+}(v)|$.

Reciprocity of Trading Interactions: As already mentioned, the value for the reciprocity between two users u and v in a directed network is 1 if there is an edge in both directions, and 0 if there is no bidirectional link between these users. Formally, we stated this feature as $T_R(u,v) = \begin{cases} 0 & \text{if } (u,v) \in E_T,\ (v,u) \notin E_T \\ 1 & \text{if } (u,v) \in E_T,\ (v,u) \in E_T \end{cases}$.

Adamic Adar: Similar to the Adamic Adar measure for the online social network, this metric could also be used for the directed trading network as $T_{AA}(u,v) = \sum_{z\ \in\ \Psi^{-}(u)\ \cap\ \Psi^{-}(v)} \frac{1}{log(|\Psi(z)^{-}|)}$.

Trading Network: Homophilic Features. All homophilic features of the trading network of the Second Life Marketplace are based on the attributes of the traded products. The attributes are category, price, and ratings of the products. We used the cosine similarity measures for the following features:

Cosine Similarity of Product Categories: To compute a value for the similarity between the product categories of a user pair (u,v), we defined two vectors $\zeta(u)$ and $\zeta(v)$. The vectors' lengths were the number of all product categories of the products u and v bought or sold. So each item i in these vectors represented a product category. The values for i were the number of products in a specific category that the user traded with. Similarly to the cosine similarity of event categories feature in the homophilic feature set of the location-based social network in Section 3.3.5, we computed the cosine similarity of product categories between u and v as $P_{CCos}(u,v) = \frac{\zeta(u)\ \cdot\ \zeta(v)}{\|\zeta(u)\|\ \|\zeta(v)\|}$.

Cosine Similarity of Product Prices: We applied the same metric for product prices. Therefore, we graduated the prices by the following scheme: $0-5L\$$, $6-10L\$$, $11-20L\$$, $21-50L\$$, $51-200L\$$,

$201 - 500L\$$, $501L\$ - \infty$. We denoted the vectors with the number of products per price step for two users u and v as $\rho(u)$ and $\rho(v)$ and so the cosine similarity of product prices between u and v could formally be written as $P_{PCos}(u, v) = \frac{\rho(u) \cdot \rho(v)}{\|\rho(u)\| \; \|\rho(v)\|}$.

Cosine Similarity of Product Ratings: We also calculated the cosine similarity for the user ratings of the products. Therefore, we classified the products in ten different rating schemes from 0.0 to 5.0 in incremental steps of 0.5. Each item i of the two vectors $\tau(u)$ and $\tau(v)$ of the users u and v represented the number of traded products by u and v in each product rating class. So we computed the value of this feature as $P_{RCos}(u, v) = \frac{\tau(u) \cdot \tau(v)}{\|\tau(u)\| \; \|\tau(v)\|}$.

Table 3.9 in this section gives a clear overview of the overall 57 used features consisting of online social network features, location-based social network features, and trading network features, each set split into topological and homophilic features.

Learning Methods and Evaluation

As mentioned in the related work before, in the literature two different kinds of learning methods are typically employed to predict links in networks: supervised and unsupervised learning methods.

Supervised Learning. The first approach we employed to predict links between seller and buyers was a machine learning approach as also referred to as supervised learning. To do so, we created a balanced dataset of user pairs with and without purchases. Therefore, we used all 5,376 user pairs that had a purchase relation in between. As negative samples we randomly chose the same amount of user pairs that had no trading interactions in between. To bring this binary classification onto a common basis, all chosen user pairs had to consist of a seller and a buyer. With this rule we prevented to select a user pair consisting of, for example, two buyers and make a purchase prediction for them, which would not have made sense. These 10,752 user pairs were split into a training set to determine characteristics of purchase interactions and a test set for verification with a tenfold cross-validation. This balanced sample of data

Table 3.9: **Overview of all Features.** This table lists the formal definitions, mean values of user pairs with ($\varnothing \mathrm{Val}_w$) and without ($\varnothing \mathrm{Val}_{wo}$) trading interactions, and significance (*** $p < .001$, ** $p < .01$, * $p < .1$). Edges for the online social network were formed based on interactions between users, while for the location-based social network co-occurrences at events and regions were used. For the trading network edges were created when a user was buying an item from another user.

	Feature	Description	Formal Definition	$\varnothing\mathrm{Val_w}$	$\varnothing\mathrm{Val_{wo}}$	Sign.
			Online Social Network			
Topological	O^+_{CN}	#Common Outgoing Neighbors	$O^+_{CN}(u,v) = \lvert\Theta^+(u) \cap \Theta^+(v)\rvert$	1.16×10^{-1}	5.00×10^{-3}	
	O^-_{CN}	#Common Incoming Neighbors	$O^-_{CN}(u,v) = \lvert\Theta^-(u) \cap \Theta^-(v)\rvert$	1.56×10^{-1}	3.96×10^{-3}	
	O^+_{JC}	Outgoing Jaccard's Coefficient	$O^+_{JC}(u,v) = \frac{\lvert\Theta^+(u) \cap \Theta^+(v)\rvert}{\lvert\Theta^+(u) \cup \Theta^+(v)\rvert}$	2.13×10^{-3}	1.66×10^{-4}	
	O^-_{JC}	Incoming Jaccard's Coefficient	$O^-_{JC}(u,v) = \frac{\lvert\Theta^-(u) \cap \Theta^-(v)\rvert}{\lvert\Theta^-(u) \cup \Theta^-(v)\rvert}$	2.39×10^{-3}	2.15×10^{-4}	
	O^+_{PS}	Preferential Attachment $+-$	$O^+_{PS}(u,v) = \lvert\Theta^+(u)\rvert \cdot \lvert\Theta^-(v)\rvert$	1.05×10^{2}	1.17×10^{1}	***
	O^-_{PS}	Preferential Attachment $-+$	$O^-_{PS}(u,v) = \lvert\Theta^-(u)\rvert \cdot \lvert\Theta^+(v)\rvert$	9.41×10^{1}	1.45×10^{1}	***
	O_R	Reciprocity of User Communication	$O_R(u,v) = \begin{cases} 0 & \text{if } (u,v) \in E_O,\ (v,u) \notin E_O \\ 1 & \text{if } (u,v) \in E_O,\ (v,u) \in E_O \end{cases}$	1.86×10^{-2}	3.72×10^{-5}	*
	O_{AA}	Adamic Adar	$O_{AA}(u,v) = \sum_{z \in \Theta^-(u) \cap \Theta^-(v)} \frac{1}{log(\lvert\Theta(z)^-\rvert)}$	1.02×10^{-1}	1.94×10^{-3}	
	O_{K001}	Katz ($\beta = 0.001$)		1.55×10^{-5}	4.06×10^{-8}	*
	O_{K01}	Katz ($\beta = 0.01$)	$O_{K\beta}(u,v) = \sum_{l=1}^{c} \beta^l \cdot \lvert path^l_{u,v}\rvert$	1.72×10^{-4}	9.58×10^{-7}	*
	O_{K1}	Katz ($\beta = 0.1$)		9.79×10^{-3}	3.08×10^{-4}	
	O_{RPR01}	Rooted PageRank ($\alpha = 0.01$)	$O_{RPR\alpha}(u,v)$ = stationary probability of v/random walk: (i) with probability $1-\alpha$ move to a random neighbor of current node, (ii) with probability α return to u	1.79×10^{-3}	1.64×10^{-5}	*
	O_{RPR05}	Rooted PageRank ($\alpha = 0.05$)		1.91×10^{-3}	1.92×10^{-5}	*
	O_{RPR15}	Rooted PageRank ($\alpha = 0.15$)		1.98×10^{-3}	2.41×10^{-5}	*
	O_{RPR3}	Rooted PageRank ($\alpha = 0.3$)		1.95×10^{-3}	3.17×10^{-5}	*
	O_{RPR5}	Rooted PageRank ($\alpha = 0.5$)		1.70×10^{-3}	4.25×10^{-5}	*
Homophilic	G_C	#Common Groups	$G_C(u,v) = \lvert\Delta(u) \cap \Delta(v)\rvert$	2.43×10^{-1}	6.06×10^{-2}	***
	G_{JC}	Jaccard's Coefficient of Groups	$G_{JC}(u,v) = \frac{\lvert\Delta(u) \cap \Delta(v)\rvert}{\lvert\Delta(u) \cup \Delta(v)\rvert}$	6.33×10^{-3}	1.25×10^{-3}	***
	I_C	#Common Interests	$I_C(u,v) = \lvert\Phi(u) \cap \Phi(v)\rvert$	2.01×10^{-2}	1.05×10^{-2}	
	I_{JC}	Jaccard's Coefficient of Interests	$I_{JC}(u,v) = \frac{\lvert\Phi(u) \cap \Phi(v)\rvert}{\lvert\Phi(u) \cup \Phi(v)\rvert}$	1.14×10^{-3}	6.51×10^{-4}	
	OI	#Interactions	$OI(u,v) = \lvert\iota(u,v)\rvert$	4.64×10^{-1}	9.30×10^{-5}	
	RR_C	#Common Check-ins	$RR_C(u,v) = \lvert\Lambda(u) \cap \Lambda(v)\rvert$	1.12×10^{-2}	6.88×10^{-4}	
	RR_{JC}	Jaccard's Coefficient of Check-ins	$RR_{JC}(u,v) = \frac{\lvert\Lambda(u) \cap \Lambda(v)\rvert}{\lvert\Lambda(u) \cup \Lambda(v)\rvert}$	2.13×10^{-4}	5.39×10^{-5}	
	RR_O	Overlap of Check-ins	$RR_O(u,v) = \frac{\lvert\Lambda(u) \cap \Lambda(v)\rvert}{\lvert\Lambda(u)\rvert + \lvert\Lambda(v)\rvert}$	1.81×10^{-4}	4.11×10^{-5}	
	RF_C	#Common Favored Regions	$RF_C(u,v) = \lvert\Xi(u) \cap \Xi(v)\rvert$	3.55×10^{-2}	3.35×10^{-3}	*
	RF_{JC}	Jaccard's Coefficient of Favored Regions	$RF_{JC}(u,v) = \frac{\lvert\Xi(u) \cap \Xi(v)\rvert}{\lvert\Xi(u) \cup \Xi(v)\rvert}$	6.50×10^{-3}	4.74×10^{-4}	*
	RF_O	Overlap of Favored Regions	$RF_O(u,v) = \frac{\lvert\Xi(u) \cap \Xi(v)\rvert}{\lvert\Xi(u)\rvert + \lvert\Xi(v)\rvert}$	5.08×10^{-3}	3.72×10^{-4}	*
			Location-Based Social Network			
Topological	L_{CN}	#Common Neighbors	$L_{CN}(u,v) = \lvert\Gamma(u) \cap \Gamma(v)\rvert$	2.57	2.38×10^{-1}	***
	L_{JC}	Jaccard's Coefficient	$L_{JC}(u,v) = \frac{\lvert\Gamma(u) \cap \Gamma(v)\rvert}{\lvert\Gamma(u) \cup \Gamma(v)\rvert}$	1.02×10^{-2}	1.02×10^{-3}	***
	L_{AA}	Adamic Adar	$L_{AA}(u,v) = \sum_{z \in \Gamma(u) \cap \Gamma(v)} \frac{1}{log(\lvert\Gamma(z)\rvert)}$	1.58	1.11×10^{-1}	***
	L_{DS}	#Days Seen	$L_{DS}(u,v) = \lvert\eta(u,v)\rvert$	3.14×10^{-1}	3.66×10^{-3}	**
	L_{MD}	Mean Distance	$L_{MD}(u,v) = \frac{1}{\lvert\omega(u,v)\rvert} \sum_{d \in \omega(u,v)} d$	4.22×10^{-1}	3.54×10^{-2}	**
	L_{K001}	Katz ($\beta = 0.001$)		3.80×10^{-5}	9.82×10^{-7}	***
	L_{K01}	Katz ($\beta = 0.01$)	$L_{K\beta}(u,v) = \sum_{l=1}^{c} \beta^l \cdot \lvert path^l_{u,v}\rvert$	8.40×10^{-4}	6.79×10^{-5}	***
	L_{K1}	Katz ($\beta = 0.1$)		2.60×10^{-1}	3.94×10^{-2}	***
	L_{RPR01}	Rooted PageRank ($\alpha = 0.01$)	$L_{RPR\alpha}(u,v)$ = stationary probability of v/random walk: (i) with probability $1-\alpha$ move to a random neighbor of current node, (ii) with probability α return to u	4.33×10^{-4}	2.82×10^{-5}	***
	L_{RPR05}	Rooted PageRank ($\alpha = 0.05$)		6.99×10^{-4}	3.15×10^{-5}	***
	L_{RPR15}	Rooted PageRank ($\alpha = 0.15$)		1.07×10^{-3}	3.92×10^{-5}	***
	L_{RPR3}	Rooted PageRank ($\alpha = 0.3$)		1.29×10^{-3}	5.07×10^{-5}	***
	L_{RPR5}	Rooted PageRank ($\alpha = 0.5$)		1.23×10^{-3}	6.64×10^{-5}	***
Homophilic	E_C	#Common Events	$E_C(u,v) = \lvert\Pi(u) \cap \Pi(v)\rvert$	3.45×10^{-1}	6.92×10^{-3}	***
	E_{JC}	Jaccard's Coefficient of Events	$E_{JC}(u,v) = \frac{\lvert\Pi(u) \cap \Pi(v)\rvert}{\lvert\Pi(u) \cup \Pi(v)\rvert}$	7.75×10^{-3}	1.78×10^{-4}	***
	E_{CCos}	Cosine Similarity of Event Categories	$E_{CCos}(u,v) = \frac{\delta(u) \cdot \delta(v)}{\lVert\delta(u)\rVert\ \lVert\delta(v)\rVert}$	5.19×10^{-1}	5.02×10^{-1}	*
	RE_C	#Common Event Regions	$RE_C(u,v) = \lvert\Upsilon(u) \cap \Upsilon(v)\rvert$	3.07×10^{-1}	1.43×10^{-1}	***
	RE_{JC}	Jaccard's Coefficient of Event Regions	$RE_{JC}(u,v) = \frac{\lvert\Upsilon(u) \cap \Upsilon(v)\rvert}{\lvert\Upsilon(u) \cup \Upsilon(v)\rvert}$	2.74×10^{-2}	1.04×10^{-2}	***
	RE_O	Overlap of Event Regions	$RE_O(u,v) = \frac{\lvert\Upsilon(u) \cap \Upsilon(v)\rvert}{\lvert\Upsilon(u)\rvert + \lvert\Upsilon(v)\rvert}$	2.19×10^{-2}	9.18×10^{-3}	***
			Trading Network			
Topological	T^+_{CN}	#Common Outgoing Neighbors	$T^+_{CN}(u,v) = \lvert\Psi^+(u) \cap \Psi^+(v)\rvert$	4.99×10^{-2}	4.95×10^{-3}	*
	T^-_{CN}	#Common Incoming Neighbors	$T^-_{CN}(u,v) = \lvert\Psi^-(u) \cap \Psi^-(v)\rvert$	1.00×10^{-1}	9.10×10^{-3}	***
	T^+_{JC}	Outgoing Jaccard's Coefficient	$T^+_{JC}(u,v) = \frac{\lvert\Psi^+(u) \cap \Psi^+(v)\rvert}{\lvert\Psi^+(u) \cup \Psi^+(v)\rvert}$	5.82×10^{-4}	1.41×10^{-4}	
	T^-_{JC}	Incoming Jaccard's Coefficient	$T^-_{JC}(u,v) = \frac{\lvert\Psi^-(u) \cap \Psi^-(v)\rvert}{\lvert\Psi^-(u) \cup \Psi^-(v)\rvert}$	3.07×10^{-3}	7.05×10^{-4}	***
	T^+_{PS}	Preferential Attachment $+-$	$T^+_{PS}(u,v) = \lvert\Psi^+(u)\rvert \cdot \lvert\Psi^-(v)\rvert$	1.69×10^{3}	4.90×10^{1}	***
	T^-_{PS}	Preferential Attachment $-+$	$T^-_{PS}(u,v) = \lvert\Psi^-(u)\rvert \cdot \lvert\Psi^+(v)\rvert$	1.51×10^{1}	9.01	***
	T_R	Reciprocity of Trading Interactions	$T_R(u,v) = \begin{cases} 0 & \text{if } (u,v) \in E_T,\ (v,u) \notin E_T \\ 1 & \text{if } (u,v) \in E_T,\ (v,u) \in E_T \end{cases}$	6.32×10^{-3}	7.44×10^{-5}	
	T_{AA}	Adamic Adar	$T_{AA}(u,v) = \sum_{z \in \Psi^-(u) \cap \Psi^-(v)} \frac{1}{log(\lvert\Psi(z)^-\rvert)}$	8.23×10^{-2}	7.34×10^{-3}	**
Homoph.	P_{CCos}	Cosine Similarity of Product Categories	$P_{CCos}(u,v) = \frac{\zeta(u) \cdot \zeta(v)}{\lVert\zeta(u)\rVert\ \lVert\zeta(v)\rVert}$	4.26×10^{-1}	1.95×10^{-1}	***
	P_{PCos}	Cosine Similarity of Product Prices	$P_{PCos}(u,v) = \frac{\rho(u) \cdot \rho(v)}{\lVert\rho(u)\rVert\ \lVert\rho(v)\rVert}$	4.71×10^{-1}	2.96×10^{-1}	***
	P_{RCos}	Cosine Similarity of Product Ratings	$P_{RCos}(u,v) = \frac{\tau(u) \cdot \tau(v)}{\lVert\tau(u)\rVert\ \lVert\tau(v)\rVert}$	6.93×10^{-1}	5.54×10^{-1}	***

resulted in a baseline of 50% for the trading prediction task when guessing at random.

We repeated all experiments ten times always choosing new random samples for the negative user pairs. As such, the results represent the averages of the respective values of the ten iterations. As a tool to run this task we chose the WEKA machine learning software [Hall et al., 2009].

As pointed out in the related work, machine learning strategies, such as decision trees, logistic regression, support-vector machines, or other types of meta-learning strategies, such as bagging or boosting, are usually used for predicting links in social networks and they work remarkably well. As such we applied a series of methods available in the WEKA machine learning framework, such as naïve Bayes, Bayes networks, decision tables, logistic regression, random forest, J48 tree, and random tree. For meta-learning we employed bagging and boosting with the same classifiers and also tested stacking. Due to space limitations, we can only present the results of a handful of approaches. The approaches we selected were random forest, logistic regression, and naïve Bayes. They showed not only the best results of the classic learning methods investigated but are also easy to implement in a real-world system. The best overall meta-learning approach was bagging with random tree, which is also included.

As an evaluation metric accuracy was chosen, as the positive and negative training examples are balanced.

Unsupervised Learning. The second approach employed to predict trading interactions was an unsupervised learning approach in the form of a user-based collaborative filtering technique, as, for example, also proposed by Liben-Nowell and Kleinberg [2007]. The intuition behind this idea was that buyers who are similar to each other will behave in a similar manner in the marketplace [Schafer et al., 2007].

We used the non-probabilistic user-based k-nearest neighbors algorithm, where for each buyer in the combined network we find their k-nearest neighbors, or more precisely, the k most similar buyers based on each individual feature and several feature sets. In a given data points collection a nearest neighbor of a query point is a data point that is closest to the query point [Beyer et al., 1999]. k defines the size of the neighborhood, for example, for $k = 10$ the 10 most similar buyers (to the given buyer) are considered based on the respective feature or feature set. Afterwards, we recommend the top-N sellers who had a trading relation with the buyers computed via k-nearest neighbors algorithm. Finally, we compare these top-N predicted sellers with the real sellers of the origin buyer.

In order to have a fair comparison between all features and feature sets, we first normalized all feature values. We applied various numbers for the parameters k and N. In this paper we only present the results of the parameters which performed best: $k = 100$ and $N = 5$.

To evaluate this approach, we used the mean average precision (MAP) [Yue et al., 2007] as performance and correctness measure, and the normalized discounted cumulative gain (nDCG) [Yilmaz et al., 2008a] as a measure for the ranking quality. We obtained the MAP by computing the mean over the average precisions (APs) from all buyers, defined as follows:

$$AP@n = \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{\#true\ sellers} \tag{3.9}$$

We denoted AP as the average precision for a buyer with $P(k)$ as the precision at cutoff k in the predicted sellers list. $rel(k)$ is 1 if the k-th seller in the list is predicted correctly, and 0 otherwise. We computed nDCG as follows:

$$nDCG@n = \frac{DCG@n}{IDCG@n} \tag{3.10}$$

with

$$DCG@n = \sum_{k=1}^{n} \frac{rel(k)}{log_2(k+1)} \tag{3.11}$$

We assume that $IDCG$ is the DCG in ideal ordering. Additionally, we report the user coverage (UC) for every used feature and feature set to show for which fraction of users these were available.

Statistical Comparison: Sellers vs. Buyers

To show the differences between user pairs with and without trading interactions we computed the mean values of the features by simply calculating the average of the feature values of the involved edges. We calculated the significance of each feature in several steps, similarly to Bischoff [2012]: First, we computed the *Levene test*—introduced by Levene [1960]—with the positive and negative edges to test for equal variances. If the *p-value* of this function was below 0.01, we calculated the *Wilcoxon rank-sum test*, otherwise the *two-sided Kolmogorov-Smirnov test*. The returning p-value was the crucial measure for the significance of a feature. Since we randomly chose the negative user pairs ten times (see section 3.3.5), we did this procedure also ten times and finally computed the mean of the values.

3.3.6 Results

This section presents the results of the implemented experiments. First, we show the mean values of the comparison between user pairs with and without trading interactions, the information gain of each feature and the accuracy value with bagging, the nDCG, the MAP, and the UC, for each of the 57 features. Finally, we present a correlation heat map of all features and the trading prediction results for different feature combinations.

Individual Features (RQ 1)

In Table 3.9 we present the mean values and significances of all features of the three networks for user pairs with ($\varnothing \mathrm{Val}_w$) and without trading interactions ($\varnothing \mathrm{Val}_{wo}$). For the online social network we observed the highest significant differences for the preferential attachment score features O^+_{PS}, O^-_{PS}. Although the values are tiny the Jaccard's coefficient of groups feature exhibits the most significant differences of all homophilic features of the online social network. For the location-based social network the features with the highest significant differences are the Katz features L_K and the common events feature E_C. For L_K user pairs with trading interactions have values about up to 38 times larger on average than user pairs without trading interactions. For E_C the differences between the user pairs are even higher. With values on average 50 times higher for user pairs with trading interactions this feature shows the most significant differences of all features used in this paper. For the topological features of the trading network we observed the largest differences between user pairs with and without trading interactions for the preferential attachment out feature T^{+}_{PS}. The cosine similarity of product categories P_{CCos} is the feature with the highest differences of all homophilic features of the trading network.

Table 3.10 provides an overview of the predictive power of each individual feature. The best performing topological online social network feature with the highest accuracy and UC was the preferential attachment score O^{-}_{PS}. With unsupervised learning the best topological feature was the reciprocity of user communication O_R. The most valuable homophilic feature of the online social network was the number of common groups feature G_C. Although the result values are quite low, for the location-based social network the best performing features were the path-based measures rooted PageRank L_{RPR} and Katz L_K, and the number of common event regions RE_C. With the best accuracy value of 0.8881 and the highest information gain, the preferential attachment feature T^{+}_{PS} of the trading network performed best.

Regarding unsupervised learning, the incoming Jaccard's coefficient T^{-}_{JC} had the highest MAP and the Adamic Adar measure T_{AA} had the best nDCG. Considering only homophilic features of the trading network, all three cosine similarity features performed quite similarly and were in the top four features regarding the information gain.

Figure 3.6 provides further insights to the question which feature is the most useful one employing information gain rank correlation analysis on all features. The colors of the bars indicate the feature group. The features with the highest information gain are again the ones from the trading network followed by location-based features.

Table 3.10: **Individual Feature Performances.** This table shows the accuracy with bagging represents the quality of each feature for predicting trading interactions with supervised learning. nDCG, MAP, and UC show the predictive power of each feature with collaborative filtering. The best results in each feature set are highlighted in bold face.

	Feature	Description	Supervised Accuracy	Unsupervised nDCG	Unsupervised MAP	Unsupervised UC
		Online Social Network				
Topological	O_{CN}^{+}	#Common Outgoing Neighbors	.5066	.0303	.0080	41.33%
	O_{CN}^{-}	#Common Incoming Neighbors	.5067	.0274	.0085	44.49%
	O_{JC}^{+}	Outgoing Jaccard's Coefficient	.5065	.0311	.0084	41.33%
	O_{JC}^{-}	Incoming Jaccard's Coefficient	.5068	.0308	.0097	44.49%
	O_{PS}^{+}	Preferential Attachment $+-$	.5441	.0468	.0101	66.52%
	O_{PS}^{-}	Preferential Attachment $-+$	**.5705**	.0428	.0143	**76.41%**
	O_R	Reciprocity of User Communication	.5091	**.0623**	**.0261**	10.46%
	O_{AA}	Adamic Adar	.5059	.0274	.0086	36.12%
	O_{K001}	Katz ($\beta = 0.001$)	.5111	.0410	.0118	44.76%
	O_{K01}	Katz ($\beta = 0.01$)	.5109	.0406	.0117	44.76%
	O_{K1}	Katz ($\beta = 0.1$)	.5111	.0404	.0118	44.76%
	O_{RPR01}	Rooted PageRank ($\alpha = 0.01$)	.5123	.0601	.0157	47.06%
	O_{RPR05}	Rooted PageRank ($\alpha = 0.05$)	.5116	.0559	.0143	47.06%
	O_{RPR15}	Rooted PageRank ($\alpha = 0.15$)	.5114	.0510	.0140	47.04%
	O_{RPR3}	Rooted PageRank ($\alpha = 0.3$)	.5121	.0484	.0135	47.02%
	O_{RPR5}	Rooted PageRank ($\alpha = 0.5$)	.5116	.0415	.0123	46.91%
Homophilic	G_C	#Common Groups	**.5333**	**.0894**	**.0336**	**75.24%**
	G_{JC}	Jaccard's Coefficient of Groups	.5327	.0874	.0332	**75.24%**
	I_C	#Common Interests	.5034	.0552	.0113	30.37%
	I_{JC}	Jaccard's Coefficient of Interests	.5025	.0573	.0122	30.37%
	OI	#Interactions	.5076	.0418	.0173	27.14%
	RR_C	#Common Check-ins	.5009	.0313	.0088	22.52%
	RR_{JC}	Jaccard's Coefficient of Check-ins	.5008	.0299	.0093	22.52%
	RR_O	Overlap of Check-ins	.5008	.0297	.0093	22.52%
	RF_C	#Common Favored Regions	.5139	.0564	.0196	73.54%
	RF_{JC}	Jaccard's Coefficient of Favored Regions	.5141	.0554	.0201	73.54%
	RF_O	Overlap of Favored Regions	.5141	.0554	.0201	73.54%
		Location-Based Social Network				
Topological	L_{CN}	#Common Neighbors	.5206	.0656	.0233	95.33%
	L_{JC}	Jaccard's Coefficient	.5202	.0641	.0227	95.33%
	L_{AA}	Adamic Adar	.5170	.0638	.0227	95.33%
	L_{DS}	#Days Seen	.5171	.0408	.0177	74.74%
	L_{MD}	Mean Distance	.5169	.0354	.0150	74.56%
	L_{K001}	Katz ($\beta = 0.001$)	.5159	**.0747**	.0187	63.88%
	L_{K01}	Katz ($\beta = 0.01$)	.5159	.0744	.0189	63.88%
	L_{K1}	Katz ($\beta = 0.1$)	.5140	.0714	.0183	63.88%
	L_{RPR01}	Rooted PageRank ($\alpha = 0.01$)	.5290	.0574	.0191	**96.96%**
	L_{RPR05}	Rooted PageRank ($\alpha = 0.05$)	.5317	.0668	.0234	**96.96%**
	L_{RPR15}	Rooted PageRank ($\alpha = 0.15$)	**.5352**	.0717	.0241	**96.96%**
	L_{RPR3}	Rooted PageRank ($\alpha = 0.3$)	.5302	.0697	.0232	**96.96%**
	L_{RPR5}	Rooted PageRank ($\alpha = 0.5$)	.5270	.0721	**.0243**	**96.96%**
Homophilic	E_C	#Common Events	.5195	.0508	.0194	93.69%
	E_{JC}	Jaccard's Coefficient of Events	.5195	.0531	.0202	93.69%
	E_{CCos}	Cosine Similarity of Event Categories	.5271	.0575	.0168	94.93%
	RE_C	#Common Event Regions	**.5346**	**.0687**	**.0239**	**99.66%**
	RE_{JC}	Jaccard's Coefficient of Event Regions	.5344	.0659	.0205	**99.66%**
	RE_O	Overlap of Event Regions	.5344	.0657	.0205	**99.66%**
		Trading Network				
Topological	T_{CN}^{+}	#Common Outgoing Neighbors	.5101	.0598	.0099	10.26%
	T_{CN}^{-}	#Common Incoming Neighbors	.5271	.9605	.7173	93.68%
	T_{JC}^{+}	Outgoing Jaccard's Coefficient	.5100	.0595	.0098	10.26%
	T_{JC}^{-}	Incoming Jaccard's Coefficient	.5271	.9660	**.7509**	93.68%
	T_{PS}^{+}	Preferential Attachment $+-$	**.8881**	.1049	.0180	13.71%
	T_{PS}^{-}	Preferential Attachment $-+$	.5351	.0271	.0057	**100%**
	T_R	Reciprocity of Trading Interactions	.5029	.2321	.0342	00.32%
	T_{AA}	Adamic Adar	.5146	**.9676**	.5549	63.30%
Homo.	P_{CCos}	Cosine Similarity of Product Categories	.7440	.2142	.0520	20.57%
	P_{PCos}	Cosine Similarity of Product Prices	.7405	**.2266**	**.0553**	15.74%
	P_{RCos}	Cosine Similarity of Product Ratings	**.7530**	.1792	.0375	**20.58%**

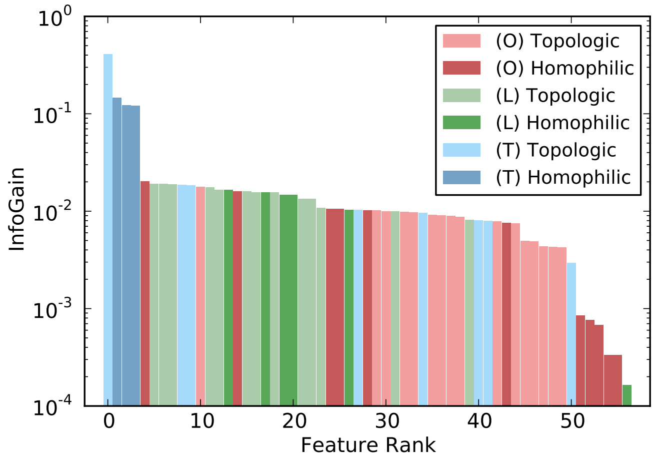


Figure 3.6: **Information Gain of Features.** This plot presents the quality ranking of the features according to their information gain. The topologic and homophilic feature sets of the online social network (O), the location-based social network (L), and the trading network (T) are color-coded. The most useful features are related to the trading network.

Feature Sets (RQ 2+3)

In predictive modeling when combining features to sets of features, it is common to take off with a correlation analysis. A correlation analysis typically helps in understanding better whether there are multi-collinearity issues which may create a problem or not later when features are combined. The correlation heat map in Figure 3.7 shows the correlations between all 57 features at a significance level at $p < .001$. For several attributes, we obtained high correlations between the Jaccard's coefficient, the Adamic Adar, the observation and the common sets for the respective attribute. Apart from these expected findings, there are high correlations between the path measures Katz and rooted PageRank in the online social network. Also the interactions feature correlate with the path measures. As expected, the features about the number of days two users have met each other, and the mean distance between two users correlate with the events features, since they have similar data bases. The idea in this paper to use additional data sources instead of just trading network features to improve the trading

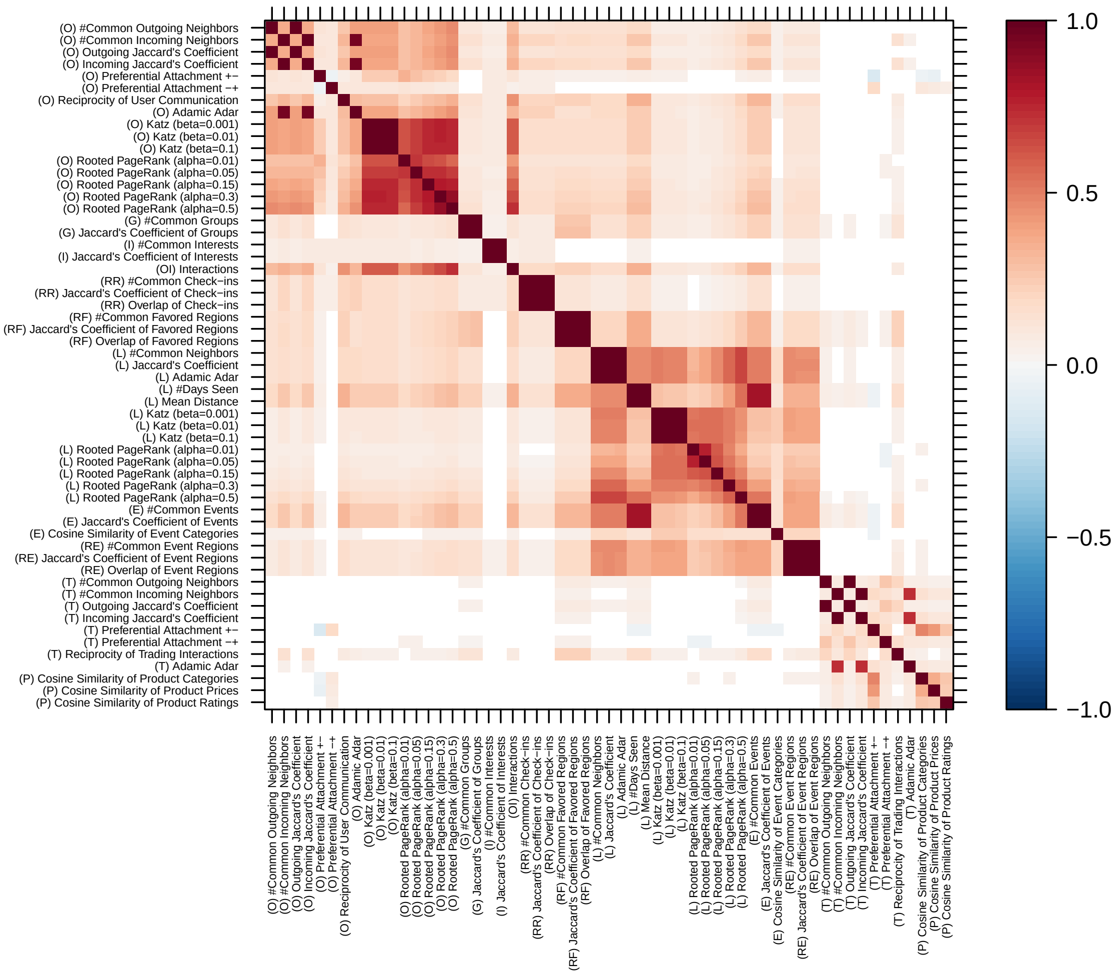


Figure 3.7: **Feature Correlations.** This plot represents a heat map indicating the Spearman feature cross-correlation values and showing the significant ($p < .001$) correlations of all 57 features.

predictions gets strengthened, since there are no significant correlations between the trading network features and features from the other data sources.

Table 3.11 provides an overview of how the several feature sets of all used data sources performed predicting trading interactions. We analyzed each feature set on its own. Using only online social network features we attained reasonable accuracy values of up to 0.6275. A bit worse performed

Table 3.11: **Feature Set Performances.** The accuracy values with random forest, logistic regression, naïve Bayes, and bagging show the predictive power of several feature sets. nDCG, MAP, and UC represent the results for the unsupervised learning approach. Best results in each set are highlighted in bold face.

Feature Set	**Random Forest**	**Logistic Regr.**	**Naïve Bayes**	**Bagging**	**nDCG**	**MAP**	**UC**	**#Feat.**
				Single Networks				
Online Social (Homo)	.5380	.5424	.5397	.5439	.0790	.0282	91.73%	11
Online Social (Topo)	.5941	.5974	.5183	.6065	.0403	.0127	100%	16
Online Social (All)	.6106	.6157	.5476	.6275	.0756	.0263	100%	27
Location-Based (Homo)	.5458	.5363	.5367	.5555	.0525	.0161	99.92%	6
Location-Based (Topo)	.5361	.5250	.5273	.5400	.0580	.0205	96.96%	13
Location-Based (All)	.5460	.5406	.5408	.5544	.0509	.0155	99.97%	19
Trading (Homo)	.7827	.6981	.7033	.8103	.1567	.0355	20.58%	3
Trading (Topo)	.8769	**.8935**	**.7722**	.8920	**.8537**	**.6200**	100%	8
Trading (All)	**.8963**	.8694	.7652	**.9119**	.6797	.5283	100%	11
				Combined Networks				
Online + Location	.6006	.5988	.5556	.6267	.0518	.0147	100%	46
Online + Trading	.9065	.8869	**.7547**	.9233	**.6440**	**.4507**	100%	38
Location + Trading	.9032	.8834	.7437	.9211	.1306	.0794	100%	30
Online + Location + Trading	**.9073**	**.8896**	.6998	**.9248**	.2493	.1374	100%	57

the combination of location-based social network features with a accuracy values of up to 0.5555. Combining the features of these two networks could not result in a performance boost, since the features of the online social network on its own are apparently quite tough for the seller-buyer prediction task. We obtained the best prediction results using features of the trading network. Using only trading network features resulted in accuracy values up to 0.9119. This astonishing result can be explained with the minor advantage the trading network has, because the trading interactions we tried to predict in our experiments originate from this network. Adding the online or the location-based social network features or both to the trading network features could slightly increase the prediction result about ≈1.5%.

Moreover, we observed that our topological features are more suitable than our homophilic features for the prediction of trading interactions as Table 3.11 also shows. This means that, by utilizing information of a network structure better trading prediction results could be obtained than by exploiting homophilic features regarding attributes of the users. The results of the collaborative filtering approach substantiate this finding.

3.3.7 Summary and Discussion

The main findings with respect to our research questions can be summarized as follows:

RQ 1. As expected, the best trading prediction results were obtained employing features from the trading network. The preferential attachment score with an accuracy value of 0.8881 and the highest information gain was the best performing feature overall by far. It also exhibited the large significant differences between user pairs with and without trading interactions. Features of the online social network and the location-based network were also to some extend useful and could achieve accuracy values up 0.5705 and 0.5352. The best individual features here were the Preferential Attachment feature in the social network and the rooted PageRank feature in the location-based network.

RQ 2. In general, the results of our experiments show that topological features are more suitable than homophilic features for the prediction of trading interactions, since the accuracy values of the topological feature sets were crucial higher than the values of the homophilic feature sets. This means that, for trading predictions the network structure is more useful than other user related attributes represented through our homophilic features. The location-based social network feature set obtained the worst prediction results of the three network sources with an accuracy value with bagging of 0.5544. To exceed the “border” of a 60% prediction probability, it was necessary to use online social network features (0.6275) or combine online and location-based social network features (0.6267). Since the trading network has a minor advantage, because the trading interactions we tried to predict in our experiments originate from this network, the best results with accuracy values up to 0.9119 could be achieved with the trading network feature sets. Adding online and/or location-based social network feature sets to them could slightly increase the prediction probability up to 0.9248. Conclusively, it could be said that online and location-based social network information on their own or in combination could result trading interaction prediction performance at an acceptable scale, which is convenient in specific setting, for example for cold-start prediction settings. As expected, the trading network information improved prediction performance strongly; it does not necessarily require the addition of further information of other network sources for trading interaction predictions. However, as noted above, trading information may not be available in all cases, therefore, the results show when the other networks can then compensate here.

RQ 3. Finally, we performed extensive experiments using both unsupervised as well as supervised learning approaches. As expected, the supervised approaches were able to score quite well, especially utilizing an ensemble classifier (i.e., bagging). Also, unsupervised strategies, which are more suitable in certain application settings where no ground truth information is available scored sufficiently.

Altogether, our results show important implications concerning personalization and recommendation approaches, as, for example, shown by Eirinaki et al. [2018]. Regarding business and managerial applications, as already has been laid out in first fundamental investigations on seller-buyer networks, for example, by Thorelli [1986] and Kranton and Minehart [2001], recommender systems in those areas play a decisive role in e-commerce [Schafer et al., 1999; Linden et al., 2003]. In particular, this specifically relates to the combined environments of digital as well as physical network structures [Liu et al., 2018], and how to effectively build recommender systems in those areas.

In the physical (real) world, for example, this relates to many systems which have a similar setup such as eBay and other online stores, where there are important links and commonalities compared to Second Life as shown in Szell et al. [2012], Lehdonvirta [2009], and Guo et al. [2011]. Then, the relations between the actors in these networks can be investigated from different (feature) perspectives—both from the physical as well as the online perspective; based on the results of this work—concerning the different features sets and their impact classification approaches can be devised, making use of the available data in the best possible and cost-efficient way in order to optimize criteria such as predictive performance, recommendation diversity, or the available number of recommendations [Herlocker et al., 2004].

Finally, explainable recommenders, thus explanation-awareness [Atzmueller and Roth-Berghofer, 2010; Nunes and Jannach, 2017; Tintarev and Masthoff, 2007] concerning the recommendations is very important, which is also enabled using the respective feature sets from multiple network perspectives. In addition, both supervised as well as unsupervised techniques can be applied here. In our experiments, random forest, logistic regression, naïve Bayes, and bagging showed the best results which makes them very good candidates for providing explanations on the recommendations; because these are "white-box classifiers", they can provide insights into the factors for inferring a certain recommendation based on the used parameters, and their weighting [Ribeiro et al., 2016; Li and Huan, 2017; Biran and Cotton, 2017].

3.3.8 Conclusions and Future Work

In our work, we collected data from three different sources of Second Life—an online social network, a location-based social network, and a trading network. Overall, we computed 57 topological and homophilic features to measure the similarities between user pairs and conducted several experiments predicting trading interactions.

In that way, we incorporated and analyzed the multiplex structure of the different networks and the individual features both from an individual as well as a collective perspective. This allowed us to identify the impact of the features from those networks both from detailed to aggregated level in order to derive actionable insights and implications from the analysis, for example for prediction, recommendation, or marketing.

As already mentioned, this paper is focused on predicting trading interactions based on features of several network sources. In the future the time component could be a very interesting factor, which was entirely neglected in this work. Time-dependent attributes could be used as prediction features or existing features could be adapted to refine the trading prediction results. When calculating a feature between two users about already traded products, for example, the products could be weighted in a way where the older trades would not be that important as newer ones. Furthermore, item to user—or in this case product to buyer—recommendations based on the existing data could also be an interesting point for future work. Here, previous work that we have presented in Lacic et al. [2015] provides a good starting point for the mentioned future analysis directions.

A further interesting direction concerns the induction of other types of features such as recently proposed in Lee et al. [2016], for example employing Latent Dirichlet Allocation [Blei et al., 2003] to find latent relations between users given their biographical information. However, more refined topic modeling approaches, such as proposed by Weng et al. [2010], would be needed in the scope of the Second Life dataset, since biographic information in the user profiles is rather sparse and typically less than 100 characters in length.

Another very interesting extension of this work would be to study the problem from a more theoretical/economical background. The study at hand does not do this extensively. Instead, we based our assumptions mostly on the link prediction and recommender systems research literature and common sociological factors indicating interactions. Building more grounded theoretical models would potentially also help to understand the nature of the problem better. So far, we have just a rough estimate. While our study suggests that predicting seller-buyer interactions from social network data is hard, Guo et al. [2011] estimates this as a moderate-hard predicting problem (at least their experimental setup and dataset suggests this). As such, more research on this problem in different kinds of datasets is needed. So far we can only claim that the problem at hand is rather easy to resolve in the Second Life dataset given trading interactions alone, while given location-based and online social network data alone, it is not.

Finally, since using the data of Second Life, the experiments in this paper were based on a virtual world. A rather important and relevant research direction concerns the understanding and modeling of digital and physical network structures and the behavior of actors therein. Therefore, an important task in the future could be to investigate how the experiments would perform if data of the real world combined with online social data were used.

3.4 Tell Me What You Want: Embedding Narratives for Movie Recommendations

This article partially answers the second research question which concerns the understanding of user preferences and narrative features that are essential in a narrative-driven recommendation scenario. Particularly, in this article, my co-authors and I set out to quantify the difficulty of the narrative-driven recommendation problem by investigating a large collection of narratives from the movie domain. We conduct the first in-depth empirical analysis of narratives from a movie suggestion board from reddit by investigating the crowdsourced dataset that we compiled in Eberhard et al. [2019b]. This dataset contains free text narratives representing movie suggestion requests from users as well as community suggestions to those requests. We ascertain the diversity of requests and suggestions as well as the usage of positive vs. negative movie features in narratives. Further, to evaluate state-of-the-art recommender algorithms for narrative-driven recommendations, we conduct a prediction experiment using embedding approaches. Specifically, we utilize document and graph embedding techniques to (i) compute algorithmic recommendations and to (ii) evaluate the importance of features extracted from requests by comparing algorithmic recommendations with the movie suggestions from the reddit community.

We find that community suggestions are oftentimes more diverse than requests, meaning that highly similar requests are frequently answered with highly diverse movie suggestions by the reddit community. Our results also suggest that users tend to write more about features that they would like to see in a movie as compared to negative features. These facts highlight the hardness of the tackled problem, making a recommendation task a challenging one. Summarizing the results of the recommendation experiment, this article uncovers that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power for narrative-driven movie recommendations on reddit. We strongly believe that a better understanding of such preferences will have strong practical implications for improving the quality of algorithmic recommendations.

Although this publication was accepted as short paper, in this thesis I present the full version of this article.

3.4.1 Abstract

Recommender systems are efficient *exploration* tools providing their users with valuable suggestions about items, such as products or movies. However, in scenarios where users have more specific ideas about what they are looking for (e.g., they provide describing narratives, such as *"Movies with minimal story, but incredible atmosphere, such as No Country for Old Men"*), traditional recommender systems struggle to provide relevant suggestions. In this paper, we study this problem by investigating a large collection of such narratives from the movie domain. We start by empirically analyzing a dataset containing free text narratives representing movie suggestion requests from reddit users as well as community suggestions to those requests. Particularly, we study the diversity of requests and suggestions as well as usage of positive vs. negative movie features (e.g., *"with happy ending"* vs. *"without happy ending"*). We find that community suggestions are frequently more diverse than requests, making a recommendation task a challenging one. Further, we find that users tend to formulate their requests *positively* and mostly write about features that they would like to see in a movie as opposed to negative features. Then, in a prediction experiment, we use embedding algorithms to assess the importance of request features including movie descriptions, genres, and plot keywords, by computing recommendations. Our findings suggest that, in our dataset, positive movies and keywords have the strongest, whereas negative movie features have the weakest predictive power. We strongly believe that our new insights into narratives for recommender systems represent an important stepping stone towards novel applications, such as interactive recommender applications.

3.4.2 Introduction

Search engines are omni-present tools designed to help users retrieve information when they specifically *know what they are looking for* (i.e., they can articulate what they want with a few simple keywords). On the other hand, users rely on recommender systems when they *are unable to specifically state what they seek* (i.e., they vaguely know what they

SUBMISSION

"**[Request] Movies about writing/writers.**
Two of my favourites are *Secret Window* and *Stranger Than Fiction*. I also liked *The Ghost Writer*. [...] I'm not a fan of horror. I know there are probably a lot of 'inspirational' movies about writing out there (I vaguely recall one with Sean Connery?). [...]"

COMMENTS

"Adaptation."

"Sean Connery movie was Finding Forrester."

⋮

Box 3.1: **Request and Suggestions Example.** In the request of this reddit submission[9] crowdworkers annotated three positive movies (i.e., *Secret Window*, *Stranger Than Fiction*, *The Ghost Writer*), a negative genre (i.e., *horror*), several positive keywords (i.e., *writing*, *writers*, *inspirational*), and a positive actor (i.e., *Sean Connery*). As suggestions from the reddit community, the crowdworkers extracted the movies *Adaptation* and *Finding Forrester* from the comments section.

want but can not articulate it). In that case, recommender systems allow users to *explore* large collections of items and find interesting items by, for example, browsing [Lamprecht et al., 2015].

Problem. In contrast to those two information seeking situations, in *narrative* scenarios users have a more specific idea what they are looking for, but the information need is often too complex to be articulated in the form of a few simple keywords. For example, there exists a vast variety of online forums for a broad range of topics where users ask peers for video/board game, movie, or music suggestions. Specifically, on such online boards they describe what they are looking for in the form of a narrative request while other users provide relevant recommendations (see Box 3.1).

Results of our recent work on such recommendation scenarios [Eberhard et al., 2019b] indicate that the problem of narrative-driven recommen-

[9] https://www.reddit.com/r/MovieSuggestions/comments/ssuhu

dations is hard and difficult to address with traditional recommender approaches. However, our research community still lacks a deeper understanding of the potential causes for the hardness of this problem. For instances, we miss insights into user preferences in narratives, and, particularly, whether users tend to illustrate their needs through (i) examples, (ii) by describing the characteristics of desired items, or (iii) by a specific combination of both examples and characteristics. Also, the question whether positively associated aspects (e.g., *writing*, *writers*, *Secret Window* and *Stranger Than Fiction* in Box 3.1) are more important for calculating recommendations than negatively associated aspects (e.g., *horror* in Box 3.1) is still unanswered in the recent research on this topic. A better understanding of such preferences will have strong practical implications for improving the quality of algorithmic recommendations.

This Work. In this paper, we set out to learn more about narratives by analyzing a movie suggestion board from reddit (r/MovieSuggestions[10]). On this board, users typically describe movies they would like to see by composing a narrative request including positive and negative movies they have already seen, genres, some key aspects (i.e., keywords), or actors that are preferred or not preferred (see Box 3.1). We start our study by empirically analyzing a dataset that consists of such narrative requests and corresponding movie suggestions from the reddit community. To quantify the difficulty of this problem we analyze the diversity of requests and their corresponding suggestions. Further, we evaluate the effects of positive vs. negative features of users on reddit. Next, we utilize document and graph embedding techniques to (i) compute algorithmic recommendations and to (ii) evaluate the importance of features extracted from requests by comparing algorithmic recommendations with the movie suggestions from the reddit community. The results of our empirical analysis reveal that community suggestions are oftentimes more diverse than requests, meaning that highly similar requests are frequently answered with highly diverse movie suggestions by the reddit community. We also find that users tend to write more about features that they would like to see in a movie as compared to negative features. The results of our recommendation experiment show that positive movies and keywords have

[10] https://www.reddit.com/r/MovieSuggestions

the strongest, whereas negative movie features the weakest predictive power for narrative-driven movie recommendations on reddit.

The key contributions of our work can be summarized as follows. We present the first detailed empirical study of narratives and feature importance from a movie suggestion board from reddit. Further, we conduct a recommendation prediction experiment to identify the most important features for generating narrative-driven movie recommendations.

3.4.3 Related Work

Narrative-Driven Recommender Systems. In context-aware recommender systems [Adomavicius and Tuzhilin, 2011] context is used for a better incorporation of current user needs in the recommendation calculation. Besides user profiles and histories, information, such as interests of a user in a specific situation [Hariri et al., 2013] or when, where, and with whom a movie was seen [Adomavicius et al., 2005], is exploited. A special context-aware recommendation scenario called narrative-driven recommendation was proposed by Bogers and Koolen [2017]. In the recommendation calculation process, besides the user history a narrative explanation of the current recommendation needs of the respective user is utilized.

In the domain of movie recommendations, Bogers [2015] investigated the discussion threads from the Internet Movie Database[11] (IMDb) message boards containing user requests for movies to watch. Besides content and metadata, such as movie descriptions or genres, the author found that searching for movies by describing their contents with narratives is essential for movie selection.

Although Glenski and Weninger [2017] showed that simple models are able to predict user interactions, such as likes, votes, clicks, and views, on reddit, recommending movies based on narrative requests on reddit exposed to be a tough problem to tackle [Eberhard et al., 2019b]. In our previous work [Eberhard et al., 2019b], we determined the suitability of well-established recommender algorithms for calculating narrative-driven movie recommendations. For evaluation, we built a crowdsourced dataset

[11] https://www.imdb.com

from reddit submissions providing narrative movie recommendation requests and comments including corresponding movie suggestions by the reddit community. The obtained results reveal that the problem of predicting narrative-driven recommendations is hard to tackle and needs further investigation to get a better understanding of narrative aspects.

Contrarily, we use this crowdsourced dataset and follow up by introducing the first in-depth empirical analysis of movie recommendation requests on reddit. Moreover, we evaluate the embedding of narratives through the document and the graph embedding techniques `doc2vec` and `node2vec`.

Embeddings Used in Recommender Systems. With the neural probabilistic language model `word2vec`, Mikolov et al. [2013a] proposed a method to learn high-quality embeddings for words in texts, where each word is mapped to a unique vector. `doc2vec` was proposed by Le and Mikolov [2014] as an enhancement of `word2vec` to allow the learning of document-level embeddings. In 2016, Grover and Leskovec [2016] proposed a graph-level embedding technique called `node2vec` that is based on random walks. As an extension to `DeepWalk` [Perozzi et al., 2014], `node2vec` diversifies the neighborhood of a node by utilizing fixed-length random walks with a mixture of breadth-first search and depth-first search schemes [Chen et al., 2019].

In the large and well-investigated research field of recommender systems and algorithms [Adomavicius et al., 2005; Bogers, 2015; Bogers and Koolen, 2017; Grbovic et al., 2015; Hariri et al., 2013], there exists a vast variety of studies based on `word2vec` and its extensions `doc2vec` and `node2vec` partly exhibiting outstanding performances [Barkan and Koenigstein, 2016; Eberhard et al., 2019b; Elsafty et al., 2018; Manotumruksa et al., 2016; Musto et al., 2015; Ozsoy, 2016; Stiebellehner et al., 2018; Chen et al., 2017, 2019; Musto et al., 2019; Kallumadi and Hsu, 2018]. Elsafty et al. [2018] showed in their job posting recommendation scenario that `doc2vec` outperforms not only `word2vec` but also the well-established content-based recommender approach TF–IDF. Chen et al. [2017] introduced a spectral clustering-based collaborative filtering recommender framework based on `node2vec`. The authors used a bipartite user-item network from a real-world dataset for their experiments and obtained results that exhibit positive effect on the improvement of baseline algorithms. In their followup work [Chen et al., 2019], they extended their approach by incorporating category information and combining multiple bipartite networks to even further improve the performance of their recommender framework. Kallumadi and Hsu [2018] evaluated the effectiveness of query-based interactive movie recommendations on IMDb data using graph-level embeddings. They created meta paths with different entities (e.g., users, movies, genres) to build movie networks as basis for their embeddings and obtained good results with `node2vec`.

In contrast to their work, we use multiple networks, each of which consists of nodes from the same type. Further, we combine `node2vec` with `doc2vec` embedding vectors based on textual movie information.

3.4.4 Empirical Analysis of Narratives

We empirically analyze the publicly available crowdsourced dataset[12] that we extracted from reddit [Baumgartner et al., 2020] in our previous work [Eberhard et al., 2019b]. The dataset contains about 1,500 narrative requests that all received at least ten suggestions and about 21,000 suggestion lists including more than 43,000 individual suggestions. We list further details of our dataset in Table 3.12. Each request in our dataset includes one or more positive movies, which are examples of movies that the user liked before. Moreover, requests frequently include additional descriptions, such as negative movies (movies that the user did not like before), positive and negative keywords describing further aspects of the movies, positive and negative genres, and finally positive and negative examples of movie actors. In Box 3.1 we show a typical example of such a request from our dataset, in which crowdworkers annotated three positive movies (i.e., *Secret Window*, *Stranger Than Fiction*, *The Ghost Writer*), one negative genre (i.e., *horror*), three positive keywords (i.e., *writing*, *writers*, *inspirational*), and one positive actor (i.e., *Sean Connery*). We also show two examples of the suggestion lists, each of them having a

[12] https://www.rbz.io/datasets

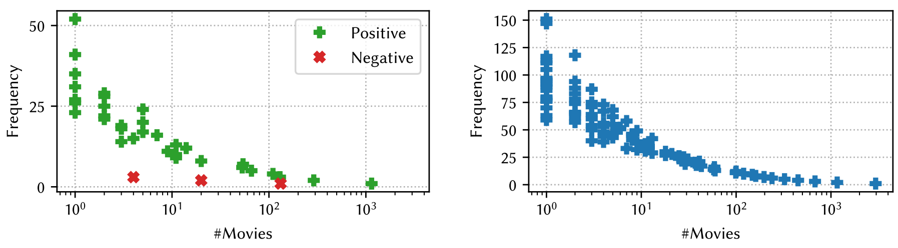
single suggestion. In the first suggestion list crowdworkers identified the movie *Adaptation* and in the second the movie *Finding Forrester*.

Dataset Characterization

Popularity Bias. We first plot the distributions of the movie occurrences in requests and suggestions in Figure 3.8. We observe a heterogeneous distributions of positive movies (see Figure 3.8a) and suggestions (see

Table 3.12: **Reddit Dataset Characteristics.** This table shows the statistics of the crowdsourced reddit dataset [Eberhard et al., 2019b].

#Requests	1,480
#Request Authors	1,244
#Movies in Requests	5,521
#Unique Movies in Requests	1,908
#Requests With Positive Movies	1,480
#Requests With Negative Movies	77
#Keywords in Requests (Without Common Words)	4,492 (3,947)
#Unique Keywords in Req. (Without Common Words)	1,878 (1,762)
#Requests With Positive Keywords	1,202
#Requests With Negative Keywords	152
#Genres in Requests	762
#Unique Genres in Requests	25
#Requests With Positive Genres	459
#Requests With Negative Genres	55
#Actors in Requests	100
#Unique Actors in Requests	79
#Requests With Positive Actors	73
#Requests With Negative Actors	7
#Suggestions	43,402
#Unique Suggestions	6,071
#Suggestion Authors	7,431
Average #Suggestions per Request	29.33
Average Duration Between Request and Suggestion	31 h 41 min



(a) Movies in Requests

(b) Suggestions

Figure 3.8: **Distributions of Movies in Requests and Suggestions.** We plot the distribution of positive (green) and negative (red) movies in requests (Figure 3.8a) and the distribution of suggestions (Figure 3.8b). Typically, users describe their request with positive movies instead of negative ones. While the majority of movies occurs in a single request, there are popular movies used across multiple requests. The most popular positive movie appears in 52 requests (3.5% of all requests). We observe similarly skewed distribution of suggestions.

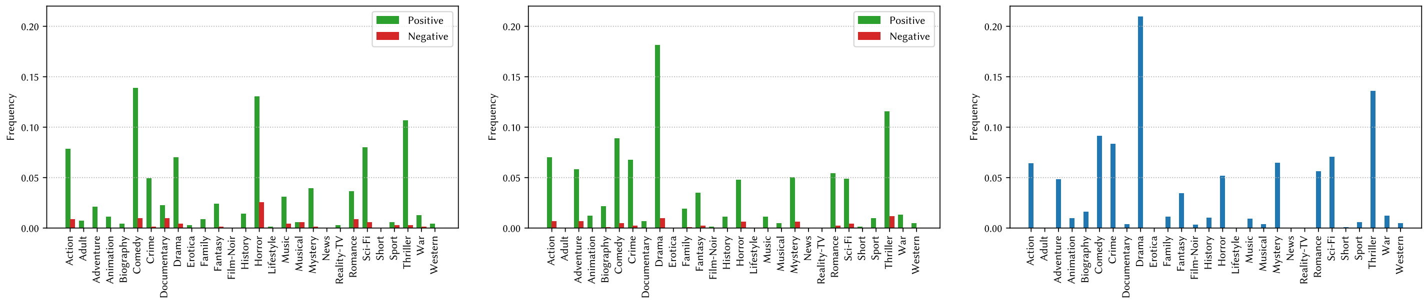
Figure 3.8b). Particularly, while the majority of the movies is mentioned only a few times, there also exist a few highly popular movies indicating a strong popularity bias. To check whether movies used as examples or suggestions are correlated we compute Spearman's coefficient. Thus, we collect all the movies occurring at least once as both positive examples as well as suggestions and compute the Spearman's rank correlation coefficient between sorted occurrence lists. We obtain $\rho = 0.638$ ($p < .001$) signaling that popular movies, such as *Fight Club* or *Memento*, are frequently used as positive examples as well as suggestions. In contrast to positive movies and suggestions, we do not observe any conclusive patterns in the distribution of negative movies due to their infrequent occurrences.

To further investigate the relations between example movies and suggestions we extract two different weighted directed graphs from our dataset. In the first graph we connect suggestions to positive movies (G^+) and in the second graph we connect suggestions to negative movies (G^-). In both graphs edge weights correspond to the suggestion frequencies. With these graphs we assess whether users tend to suggest popular movies to popular positive movies in requests, and unpopular movies to unpopular ones, by calculating correlation of degrees over the links (i.e., degree assortativity r [Newman, 2003]). For each node we aggregate in- and out-degree and obtain $r = 0.044$ for G^+ and $r = -0.018$ for G^- suggesting uncorrelated node degrees. In other words, popular movies are suggested in response to both other popular movies as well as unpopular ones.

Hence, our initial findings indicate that automatic recommender algorithms trained on this or similar data may be strongly influenced by the popularity bias and should follow advanced strategies to account for this bias in order to, for example, increase beyond-accuracy metrics, such as diversity or novelty.

Weak Agreement in Movie Genres. In the next step, we plot the positive and negative genre distributions (i.e., the genres directly mentioned in the user requests) in Figure 3.9a, the distributions of genres of movie examples in Figure 3.9b, and genres of suggestions in Figure 3.9c, the latter two extracted from a set of 28 genres from IMDb.

In our dataset, the most frequent positive genres in requests are *Comedy* (99 occurrences), *Horror* (93), and *Thriller* (76). The genre with most negative occurrences is *Horror* (18 occurrences). In case of the genres of example movies, the most frequent ones are *Drama* movies (1,122) followed by *Thriller* (715) and *Comedy* (550). The most frequent suggestion genres are also *Drama* (in 27,559 suggestions), *Thriller* (17,848), and *Comedy* (12,036) indicating an overlap between genres in requests and suggestions. We quantify this overlap by computing genre assortativity of the G^+ and G^- graphs we constructed previously. Particularly, the genre assortativity measures the correlation of the genres of example movies and the genres of the corresponding suggestions. We obtain positive assortativity coefficients for both graphs G^+ ($r = 0.162$) and G^- ($r = 0.191$) signaling a weak overlap between movie genres in requests and suggestions. While we expect to observe such an overlap between suggestions and positive movies, an overlap between genres of negative movies and suggestions is surprising. We hypothesize that negative movies are often used as examples of movies that users have already seen, asking the community to provide alternative suggestions. Thus, we manually inspect all requests with negative movies finding a substantial number of requests providing ev-



(a) Genres in Requests

(b) Genres of Movies in Requests

(c) Genres of Suggestions

Figure 3.9: **Frequency of Genres.** These plots represent the occurrences of positive and negative genres in requests (Figure 3.9a), genres of positive and negative movies in requests (Figure 3.9b), and genres of suggestions (Figure 3.9c). We find that *Comedy*, *Horror*, and *Thriller* are the most frequent genres in requests, whereas *Drama* and *Thriller* are the most frequent genres of positive movies and suggestions. We also observe a weak correlation between movie genres in requests and suggestions.

(a) Positive Keywords. (b) Negative Keywords.

Figure 3.10: **Word Clouds of Positive and Negative Keywords.** These figures show word clouds of positive (Figure 3.10a) and negative (Figure 3.10a) keywords in requests. The size of a word reflects the frequency over all positive or all negative mentions of keywords, respectively.

idence for our hypothesis (e.g., "*Looking for third for a three-way with Her and Ex Machina [...] And I don't really want to see A.I. again. [...]*").[13] We also find examples of requests in which negative movies are used to exclude otherwise matching suggestions, for example, due to the personal animosity of the requesting user (e.g., "*Scariest Horror Movies ? [...] I love movies like Annabelle and The women in black. But I don't like The Ring [...]*"[14]).

Users Prefer Dark over Hollywood Movies. In Figures 3.10a and 3.10b we depict word clouds of positive and negative keywords after removing stopwords. In each word cloud, the size of a keyword corresponds to its relative occurrence frequency among positive or negative keywords. The word clouds suggest that users on reddit prefer to watch *dark* movies in some kind of *psychological* setting (see Figure 3.10a) rather than *hollywood* or *gory zombie* movies (see Figure 3.10b). These findings could potentially explain the larger fraction of *Drama*, *Thriller*, and *Mystery* movie suggestions as compared to the fraction of such movies in requests (see Figures 3.9b and 3.9c).

[13] https://www.reddit.com/r/MovieSuggestions/comments/5x7kym
[14] https://www.reddit.com/r/MovieSuggestions/comments/393ue3

Next, we investigate whether the semantic of the positive and negative keywords describes distinctive movie properties. To that end, we assess whether they come from different topical clusters by representing the words with English word vectors from fasttext[15] pre-trained on Common Crawl and Wikipedia. The vectors provide contextual embeddings of words in a high-dimensional space, in which words commonly used in a similar context appear spatially closer together [Grave et al., 2018]. After extracting keyword vectors we reduce the space dimensions to three dimensions by t-SNE [Maaten and Hinton, 2008]. The visual inspection of the t-SNE plots does not reveal clusters of neither positive nor negative keywords. Hence, while the positive and negative keywords are not semantically separated, their interactions with other features, such as movie examples or genres, possibly provide useful information for the suggesting users. We investigate this differentiating potential of keywords later in more detail.

Finally, similarly to genres we determine whether plot keywords (which we extract from IMDb for each movie in our dataset) of example movies in requests are correlated with plot keywords in suggestions. Plot keywords represent the gist of the storyline of a movie in a couple of concise terms or short phrases. We again compute the assortativity coefficient r for graphs G^+ and G^- this time taking plot keywords as node attributes. We obtain coefficients close to zero for both graphs G^+ ($r = 0.008$) and G^- ($r = 0.007$) indicating uncorrelated plot keywords of example movies and suggestions.

Summary. Our initial dataset characterization suggests a strong popularity bias towards movies popular in this particular reddit community. The fact that there is a significant correlation between positive movies and suggestions also indicates that users frequently describe similar requests with differing movie examples. This highlights the importance of context in such narrative-driven recommendation requests as we observe only weakly correlated genres and no distinguishing use of positive and negative keywords. Thus, users signal their recommendation needs through specific

[15] https://fasttext.cc/docs/en/crawl-vectors.html

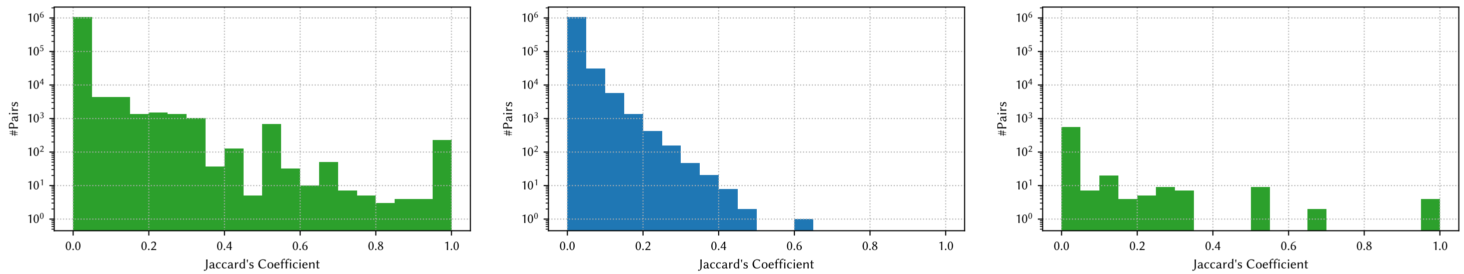
and distinctive combinations of various features including example movies, genres, or keywords.

Suggestions Diversity

We continue our empirical analysis by investigating how diverse community suggestions are, which we see as an important next step in characterizing the narrative-driven recommendation problem and the hardness of this problem. Particularly, as reddit submissions resemble a typical discussion board structure, we expect that each following user suggesting movies recommends movies different to the already suggested ones. Moreover, as users can browse the submission lists and corresponding suggestions from the past, we also expect a high suggestion diversity across different requests. Taken all together, we hypothesize that highly diverse community responses will render narrative-driven recommendations as a challenging task for automatic recommender algorithms.

Overlapping Examples Result in Differing Suggestions. In the first step, we investigate how overlap in positive movie examples across requests relates to overlap (or lack thereof) in suggestions. Thus, we compute a standard overlap measure, the Jaccard's coefficient, of positive movies from all pairs of requests and corresponding suggestions. We plot all overlap distributions in Figure 3.11.

We find that both requests and suggestions are typically dissimilar to the majority of other requests with average of 0.003 (sample standard deviation $s = 0.028$) and suggestions with average of 0.008 ($s = 0.019$). However, while there is still a substantial number of requests highly similar to at least one other request (e.g., with similarities ≥ 0.5), we do not observe such similarities in pairs of suggestions. Particularly, the similarity distribution of request pairs has a significantly longer tail than the similarity distribution of suggestions (see Figures 3.11a and 3.11b). For example, we observe the maximal request similarity of 1 (i.e., identical positive movies in a request pair) in 230 request pairs comprising 233 different requests, which constitute more than 15% of all requests. These numbers increase to 246 request pairs comprising 245 requests (16% of all



(a) Positive Movies

(b) Suggestions

(c) Positive Keywords (Positive Movies Similarity ≥ 0.5)

Figure 3.11: **Overlap of Movies and Keywords in Requests and Suggestions.** These figures (y axes on the log scale) show the Jaccard's coefficient of positive movies (Figure 3.11a), suggestions (Figure 3.11b), and positive keywords (Figure 3.11c), with positive movies similarity ≥ 0.5 for all request pairs. We observe positive movies and positive keywords in requests having a longer tail with a higher probability for high similarity values as compared to suggestions. For example, we find 1043 pairs consisting of 622 (42.03%) requests with positive movies, 708 pairs consisting of 410 (34.11%) requests with positive keywords, and only a single suggestion pair with Jaccard's coefficients ≥ 0.5. These findings indicate a strong diversity of suggestions in our dataset.

requests) and 1,043 pairs coming from 622 requests (42% of all requests) for similarity values ≥ 0.75 and ≥ 0.5, respectively. On the other hand, we observe a single pair of suggestions with a similarity ≥ 0.5. Moreover, the average similarity of the corresponding suggestions for the request pairs with similarities ≥ 0.5 is as low as 0.056 ($s = 0.061$).

Varying Tastes Lead to Suggestion Diversity. One possible explanation for the diversity in suggestions may be that suggestions are provided by different users with diverging movie preferences. As we expect that users will follow their own tastes and preferences while suggesting movies, the varying user preferences will result in different suggestions even to identical requests.

Thus, we start by removing all suggestion lists from deleted users (3% of suggestion lists), who are denoted as "[deleted]" in our dataset. We find that the remaining users create on average only 2.74 suggestion lists ($s = 7.27$), which may lead to increased diversity in the suggestions. In fact, we find that there are only 260 users that provide more than ten suggestion lists while the most of the users (4,362) only provide a single suggestion list. Previous studies showed that the online communities are highly dynamic with low rates of user retention [Yang et al., 2010; Suh et al., 2009] potentially providing an explanation for this finding. Next, we analyze the overlap of users for requests with identical positive movies. In 230 of such request pairs we find an average overlap (as measured with the Jaccard's coefficient) between suggesting users to be 0.010 ($s = 0.024$). Similarly, the average overlap of suggesting users for the request pairs with similarities ≥ 0.75 and ≥ 0.5 is as low as 0.010 ($s = 0.024$) and 0.008 ($s = 0.020$), respectively. Thus, this observation corroborates the previous finding suggesting that typical users respond sporadically by answering only to a small number of posted requests or by providing only a small number of suggestions. Finally, to assess the effect of user personal preferences on the suggestions that they make we need first to eliminate the effects of confounding factors, such as additional positive keywords or genres. Therefore, we control for these features and extract a subset of requests with identical positive movies and no additional information. We find 14 such pairs comprising 17 requests (1% of all requests). Among those request pairs we find no overlap between responding users. The average similarity of 0.081 ($s = 0.050$) for suggestions between these pairs aligns with our hypothesis about the importance of user preferences when providing suggestions. However, we retain from making any conclusions based on these results due to a small number of such request pairs in our dataset.

Community Responses Take Long Time. Further, the time span between the posting of requests and the suggestions from the community is another possible factor leading to diversity in responses. We find that for the request pairs with positive movie similarities ≥ 0.5, the average time span between the requests is about one year and eight months with sample standard deviation being about one year and four months (we obtain comparable results for similarity thresholds of 0.75 and 1). As numerous new movies are released in such prolonged periods of time, newer suggestions possibly also include newly released movies. This result indicates that the community responses tend to be up-to-date and that the recency may be an important factor when providing narrative-driven recommendations.

Differentiating Between Requests

Our findings so far confirm our intuition that community responses are highly diverse, at least with respect to highly overlapping positive movie examples. This constitutes a further evidence hinting towards the importance of the context provided by a given narrative as well as the importance of the additional information that users provide in their requests. In the next step of our empirical analysis, we investigate such additional information in more detail. To that end, we continue by analyzing positive keywords in requests.

Keywords Differentiate Between Similar Requests. In analogy to positive movies we compute Jaccard's coefficient between positive keywords for all pairs of requests. Similarly to positive movies and suggestions, we find that, on average, positive keywords are highly dissimilar to each other with average of 0.004 ($s = 0.031$). To further investigate whether the positive keywords are able to discriminate between requests with highly similar positive movies (similarities ≥ 0.5), we look more closely on the distribution of the keyword similarities (see Figure 3.11c) for this subset. From those pairs we extract 619 pairs with both requests including positive keywords (457 requests in total). The average keyword similarity for this selection of request pairs is 0.031 ($s = 0.118$).

While these initial results indicate that keywords play an important role in distinguishing the user request between otherwise highly similar requests, they may be distorted by additional information included in those requests, such as negative movies, negative keywords, positive and negative genres, or positive and negative actors. Thus, we once more extract a subset of requests with highly similar positive movies (i.e., with similarities ≥ 0.5) but that include only positive keywords as the additional information. This leaves us with 245 request pairs comprising 224 such requests. We find a similar distribution of keyword similarities for those requests as in Figure 3.11c. The average of this distribution is 0.032 ($s = 0.127$), thus corroborating our previous results on the distinctive role of the keywords in recommendation requests. Moreover, we find that the average similarity of the suggestions corresponding to this request subset is low at 0.061 ($s = 0.067$), hinting once more at differentiation potential of positive keywords. Note that our results are robust with regard to varying thresholds of positive movies similarity. Particularly, we repeat the same calculations for positive movie similarities ≥ 0.75 and 1 and obtain comparable results.

Thus, our analysis suggests that users frequently use overlapping positive movies but differing keywords to further refine their requests, which highlights the importance of keywords for the community when suggesting movies (see Box 3.2). Taken all together, we argue that the problem of narrative-driven movie recommendations is not as simple as "filtering based on mentioned movies" but rather requires the inclusion of further key aspects from requests, such as keywords. Finally, please note that we repeat the same analysis with positive genres and actors, but due to the low number of requests including such additional information and having the positive movie similarity ≥ 0.5 (4 pairs with 8 requests for genres and no pair for actors), we do not obtain any conclusive results and thus do not report them here.

SUBMISSION[16]

"Looking for good depressing movies
I'm looking a well made and entertaining depressing movie. I already have The Road on my list so anything similar to that would be cool."

SUBMISSION[17]

"Looking for post apocalyptic/survivalist movies?
Something along the lines like The Road or even anything involving zombies. International film recommendations are welcome. Many thanks in advance."

SUBMISSION[18]

"[REQUEST] Movie to bring me back to life after 'The Road'
Seriously this movie is bumming me the fuck out, I need something to make me laugh."

Box 3.2: **Request Examples.** Those three requests contain one identical positive movie (i.e., *The Road*), but differing positive keywords (e.g., *entertaining depressing*, *post apocalyptic*, or *make me laugh*). The Jaccard's coefficients of the corresponding suggestions to the requests are zero, meaning that there is not a single overlap between the suggestions.

3.4.5 Feature Importance

In this chapter, we complement our empirical analysis with evaluation of feature importance within a practical application of narrative requests from reddit in a recommender system. Specifically, we use the state-of-the-art document and graph embedding methodologies `doc2vec` and `node2vec`, which position movies with similar characteristics close-by in the resulting vector space. To train our models we use the publicly available IMDb dataset (see Table 3.13) and supplement it with user reviews we collect for all movies on IMDb. Note that we use `doc2vec` for movie descriptions and user reviews, while we construct graphs based on overlaps of characteristics between movies (e.g., common genres, plot keywords, or user

[16] https://www.reddit.com/r/MovieSuggestions/comments/r9z6l
[17] https://www.reddit.com/r/MovieSuggestions/comments/6dqnj2
[18] https://www.reddit.com/r/MovieSuggestions/comments/1fg4al

ratings) to create vector embeddings using `node2vec`. Finally, to generate recommendations, we calculate similarity between the embedding vectors of all movies and the embeddings of the features provided in the reddit requests and select the most similar ones. In a preprocessing step, we remove all stopwords from movie descriptions and user reviews. Further, we only keep movies with at least 1,000 user ratings, at least one user review, a movie description, and at least one person in the cast, to account for the data sparsity.

Table 3.14 lists the characteristics of all graphs for `node2vec`. Note that we scale edge weights and filter edges via thresholds for some of the resulting graphs (cf. individual paragraphs) to account for noise from rarely occurring movie characteristics or to smooth the weights by downscaling. In turn, this decreases the number of (reachable) nodes in these graphs. While we are aware that some movies might not be represented in some

Table 3.13: **IMDb Dataset Characteristics.** This table shows the statistics of the IMDb data that we used for our experiments.

#Movies	11,578
Movie Descriptions and User Reviews:	
#Words	250,928,734
#Unique Words	539,367
Average #Words per Movie	$\approx$ 21,672.89
#Ratings	144,021,151
Average #Ratings per Movie	$\approx$ 12,439.21
#Genres	32,767
#Unique Genres	25
Average #Genres per Movie	$\approx$ 2.83
#Plot Keywords	1,124,510
#Unique Plot Keywords	89,003
Average #Plot Keywords per Movie	$\approx$ 97.12
#Credits	667,279
#People in Casts and Crews	322,881
#Actors	294,533
Average #Actors per Movie	$\approx$ 25.44

of the graphs anymore (cf. S in Table 3.14), we still are able to keep all movies from the dataset when combining embeddings.

Source and Training of Embeddings

Movie Descriptions and User Reviews from IMDb (`doc2vec`). We use textual data about movies collected from IMDb for all movies in the dataset. We leverage their descriptions containing plot summaries, synopses as well as user reviews to train the `doc2vec` embeddings. Eventually, every movie is represented as a vector based on its textual information.

User Ratings from IMDb (`node2vec`). We use user ratings from IMDb as popularity proxy and create an undirected ratings graph where nodes represent movies and weighted edges between nodes indicate how many users rated both movies. We use `node2vec` to compute embeddings for each movie in the graph. We smooth the edge weights by dividing them with 2,500 without remainder resulting in a network density of approximately 25%.

Genres from IMDb (`node2vec`). We use movie genres from IMDb to create a genres graph for all movies in our dataset. Nodes represent movies and weighted edges between nodes state the number of common genres two

Table 3.14: **Graph Statistics.** This table lists the characteristics of the created graphs based on IMDb data. Aside from the number of nodes (n), edges (m), and the graph type (Type; directed/undirected) we report the average degree ($\overline{k}$), the number of connected components (c), and the fraction of the largest connected component (S).

Graph	n	m	**Type**	$\overline{k}$	c	S
User Ratings	6,424	5.54m	Undir.	1,725	2	0.997
Genres	11,578	9.83m	Dir.	849	1	1.0
Plot Keywords	8,529	4.43m	Undir.	1,040	5	0.999
Casts and Crews	11,550	1.50m	Undir.	260	3	0.999
Years	11,576	8.34m	Undir.	1,441	3	0.999

movies share. We use `node2vec` to compute embeddings for each movie in the graph. To reduce noise we only keep edges between movie pairs, if the number of mutual genres is greater than 50% of the number of genres of the first movie, resulting in a directed genre graph.

Plot Keywords from IMDb (`node2vec`). We use movie plot keywords to create an undirected graph, with nodes representing movies and weighted edges the number of plot keywords two movies share. We downscale the edge weights by dividing them with 12 without remainder resulting in a network density of 12%. We then compute `node2vec` embeddings for movies in the graph.

Casts and Crews from IMDb (`node2vec`). We create an undirected graph based on the casts and crews of the movies in our dataset. Nodes represent movies and a weighted edge between two nodes represents the number of people involved in both movies. We use `node2vec` to compute movie embeddings for this graph.

Years from IMDb (`node2vec`). To model the similarity of movies based on their release time, we build an undirected graph with movies as nodes and edges connecting two movies released within a period of two years. We set the edge weight to 2, if movies are released in the same year, and to 1 otherwise. We use `node2vec` to compute embeddings for each movie in the graph.

Model Training and Hyperparamter Optimization. To train and optimize our models we use both the reddit dataset and the IMDb dataset. First, we create a validation and a test set by chronologically splitting the reddit dataset (80%/20%). Second, we train our embeddings on IMDb data and use the reddit data from the validation set to execute an exhaustive grid search experiment for optimizing hyperparameters for the `doc2vec` and `node2vec` models. Particularly, we evaluate vector dimension (Dim), window size (WS), number of iterations over the corpus (#Iter), learning rate (LR), minimum learning rate (MinLR; value to which the learning rate linearly drops as training progresses), threshold to remove words with a lower total frequency (MinFreq), value for negative sampling specifying how many "noise words" should be drawn (NS), value for the exponent used to shape the negative sampling distribution (NSExp), training algorithm, random walk length (WL), and number of walks (#Walks). We list all the parameter intervals in Table 3.15 and report the parameter configurations with the best prediction results on the validation set in Table 3.16.

Calculating Movie Recommendations

After our models are trained, we calculate similarity between movie embeddings and the requests. Specifically, we use request features for our recommendation calculations as follows.

Movies. For each `doc2vec` and `node2vec` model (i.e., movie descriptions and user reviews, user ratings, genres, plot keywords, casts and crews, and years), we compute the cosine similarity of the embedding vector of a given positive movie to all other movie embeddings from the same model. In case that multiple positive movies are given in one request, we sum up the computed cosine similarities for each movie in our dataset, and subtract similarities for all given negative movies.

Genres. To include genres in the computation of recommendations we first represent each genre from a given request by its corresponding word

Table 3.15: **Hyperparameters.** This table shows the applied values for hyperparameter optimization.

Hyperparameter	**Tested Values**
Dim	$\{100, 200, 300, ..., 1{,}000\}$
WS	$\{1, 2, 3, ..., 10, 15, 20, 50, 100\}$
#Iter	$\{5, 10, 15\}$
LR	$\{0.001, 0.01, 0.025, 0.05, 0.1\}$
MinLR	$\{0.0001, 0.001, 0.01, 0.1\}$
MinFreq	$\{0, 1, 5, 10, 100, 500, 1{,}000, 5{,}000, 10{,}000\}$
NS	$\{0, 5, 10, 20\}$
NSExp	$\{0, 0.25, 0.5, 0.75, 1\}$
Algorithm	{Distributed Memory (`PV-DM`), Bag of Words (`PV-DBOW`)}
WL	$\{100, 200, 300, ..., 900\}$
#Walks	$\{5, 10, 100, 500\}$

Table 3.16: **Best Parameter Configurations.** This table shows the best performing parameter configurations of the `doc2vec` and `node2vec` models determined via hyperparameter optimization.

	Type	**Dim**	**WS**	**#Iter**	**LR**	**MinLR**	**MinFreq**	**NS**	**NSExp**	**Algorithm**	**WL**	**#Walks**
`doc2vec`	Movie Descr. and User Reviews	400	1	10	0.05	0.001	5,000	5	0.5	`PV-DBOW`	—	—
	Keywords in Requests	500	1	10	0.05	0.0001	5,000	5	0.5	`PV-DBOW`	—	—
	Genres in Requests	500	10	10	0.001	0.001	5,000	5	0.5	`PV-DBOW`	—	—
`node2vec`	User Ratings	100	5	15	—	—	—	—	—	—	900	100
	Genres	100	5	10	—	—	—	—	—	—	900	100
	Plot Keywords	100	10	10	—	—	—	—	—	—	900	100
	Casts and Crews	200	10	10	—	—	—	—	—	—	200	100
	Years	200	5	10	—	—	—	—	—	—	900	100

vector from the doc2vec model. We add up all word vectors for positive genres and subtract all vectors for negative genres, if provided. Finally, we compute the cosine similarities between the resulting vector and all movie vectors from the doc2vec model.

Keywords. We use all positive keywords from requests as input for our doc2vec model to obtain a vector representing positive keywords. If negative keywords are provided, we use them to obtain another vector representing negative keywords, and subtract it from the positive keywords vector. Finally, we compute the cosine similarities from the resulting vector to all movie embeddings from the doc2vec model.

Actors. If positive actors are mentioned in a request we treat them as keywords and add their names to all given keywords to be inferred in our doc2vec model. If negative actors are given we remove all movies with these actors in the cast, comparable to our previous work [Eberhard et al., 2019b].

Generating and Evaluating Recommendations. Before we generate recommendations we first filter out predecessors and successors for movies specified in the request, as we assume that users do not want to receive a list of movie series they are already familiar with [Eberhard et al., 2019b]. Further, we consider the time (i.e., the year) a request was created and remove movies that were released later. After these filtering steps, we then evaluate the importance of positive as compared to negative mentioned features in the requests as well as the importance of all features individually in isolation of other features. Finally, we linearly combine cosine similarities of individual models into a single similarity measure for each movie from our dataset. To evaluate the importance of features we treat the coefficients of the linear combination of our models as parameters that we once more optimize via exhaustive grid search. Thus, to obtain the optimal weight for each of the eight models, we iterate over 21 different weights for each model ranging from 0 to 1 starting with steps of 0.2. We then take more granular steps of 0.1 and 0.05 in regions where we achieve promising results, resulting in inspection of about 1.7 million configurations in total.

In each experiment, we evaluate the performance of our recommender approach on the test set. For evaluation, we limit our final recommendation lists to ten movies[19] and compute precision, recall, and F1 score [Powers, 2011; Eberhard et al., 2019b] by comparing our recommendations to the suggestions of the reddit community. As sanity check of our approach we implement a most popular baseline by always using the ten most occurring movie suggestions from the validation set. As additional baselines for our approaches we use the results obtained in our previous study [Eberhard et al., 2019b]. Note that the target of this paper is not to beat every existing state-of-the-art approach tackling the narrative-driven recommendation problem but rather to analyze the importance of user preferences and request features.

Results

Positive vs. Negative Information. To measure the impact of explicitly mentioned negative features, we first conduct our experiments leveraging only positive information including negative information only in the second iteration of our experiments. Our results reveal that additionally considering negative features in requests in most cases leads to slightly worse results than with positive features only. Every model, except years and casts and crews, achieves the best F1 scores when considering only positive features from requests suggesting that recommender algorithms can potentially ignore negative information, which is also in line with the results of our empirical analysis.

Individual Features. We obtain the highest F1 score of 0.086 with the `doc2vec` model followed by the graph-based `node2vec` models with user ratings (0.039) and plot keywords (0.034), and the `doc2vec` model with inferred keywords from requests (0.034). One possible reason for the increased importance of movie descriptions and user reviews might be that the `doc2vec` model is trained on a large amount of textual information, potentially capturing additional latent attributes that are missing

[19] Recall@10 and F1 score@10 have mean upper limits of 0.45 and 0.60 respectively, as the average number of movie suggestions from the community per request is 29.22 in the test set.

in the ratings and plot keywords that are used to train the respective `node2vec` models. We achieve the lowest F1 scores with the graph-based genres (0.016) and years (0.002) models with IMDb data. Note that all `node2vec` models only use the given movies as input without considering further information provided in a request, such as keywords or genres.

In summary, our results with individual features indicate that narrative-driven movie recommendations exhibit only minimal overlaps with human suggestions. As each model only covers a specific feature (e.g., movies, keywords, or genres) of the requests, the computed recommendations do not fully match suggestions from humans having all request information available. The most important feature individually are positive movies as the result of the `doc2vec` model shows. Additionally, the importance of keywords is in line with the results of our empirical analysis, where we found out that keywords are important to discern between otherwise similar requests.

Feature Combinations. In Figure 3.12 we depict the feature weight of each model for the best performing parameter configuration of the linear combination of the models. We find that movie descriptions and user reviews are most important with a weight of 0.9, followed by keywords in requests with a weight of 0.8. All other models exhibit comparably low weights, with the years and casts and crews models being completely ignored with weights of 0. Using this weighted model combination we obtain an overall F1 score@10 of 0.115.

In summary, the results show that our approach (F1 score@10 of 0.115) outperforms the most popular baseline (F1 score@10 of 0.039) and is comparable with the state-of-the-art approaches from our previous work [Eberhard et al., 2019b]. It almost reaches the performance of our best post-filtering approach using `doc2vec` (F1 score@10 of 0.126).

Empirical Results vs. Prediction Results. Somewhat low F1 values that we achieve with our prediction experiments are in line with the results of our empirical analysis showing that overlapping movie examples oftentimes result in differing suggestions providing further evidence for the difficulty of the narrative-driven recommendation problem. Further, in our

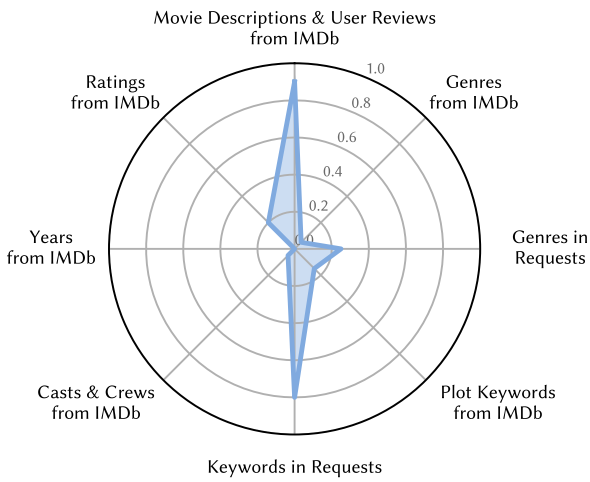


Figure 3.12: **Feature Importance.** We determine the importance of features in our requests through hyperparameter optimization of our models. The obtained weight for each model corroborates the findings of our empirical analysis. The movie feature based on textual movie information and user reviews as well as keywords in requests exhibit to have the strongest predictive power in case of movie recommendations on reddit.

prediction experiments the features with the highest weights are the movie feature based on movie descriptions and user reviews, and the keywords in requests. This finding corroborates the results of our empirical analysis, suggesting that the most important narrative aspects for recommendations are positive movies combined with keywords further describing the user needs. Moreover, a high importance of the user review feature is a further strong evidence for the popularity bias as we expect that movies popular on reddit also collect more reviews on IMDb.

3.4.6 Conclusions and Future Work

In this paper, we evaluated the importance of different features extracted from requests for movie recommendations and analyzed the effects of positive vs. negative examples. We conducted an empirical analysis of a crowdsourced reddit dataset consisting of movie recommendation requests in the form of narratives and movie suggestions in the form of comments by the community. We found that users mainly focus their requests on positive aspects, such as movies they liked or keywords that describe the movies they would like to see. The fact that similar requests frequently yield diverse suggestions indicates that implementing automatic narrative-driven movie recommendations is a hard problem.

To compute and evaluate movie recommendations based on narrative requests we used `doc2vec` and `node2vec` trained on textual information of movies from IMDb. We employed the requests from the reddit dataset to compute movie recommendations through `doc2vec` and `node2vec` and compared them with the movie suggestions from the reddit community. We presented an analysis of the importance of features as well as the role of positive and negative aspects provided in requests. Our results showed that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power in the case of movie recommendations on reddit.

For future work, we plan to supplement existing approaches by extracting (e.g., by using natural language processing techniques) and incorporating information from reddit requests already in the training phase of the embeddings. For example, as our results suggest, positive keywords are important for distinguishing between requests and their automatic extraction (e.g., in the form of named entities) should be considered in the training phase. Furthermore, we aim on investigating other embedding techniques, such as `item2vec` or `community2vec` to possibly achieve a further performance improvement. Next, we will analyze and evaluate other domains, such as suggestion boards for books, board games, or video games. For example, we plan to compare the characteristics of different communities and to investigate which aspects are important in which community to provide relevant recommendations. Finally, we plan to implement a recommender bot that users can query for recommendations while providing requests. Using this bot, we will conduct experiments to evaluate the importance of additional metrics, such as diversity or novelty, in the context of narrative-driven recommendations.

3.5 Evaluating Narrative-Driven Movie Recommendations on Reddit

This article completes the answer to the second research question by concerning the suitability of state-of-the-art recommender approaches to approximate human suggestions in a narrative-driven recommendation scenario. To that end, my co-authors and I compile and publish a reference dataset by collecting and parsing narrative requirements from users on reddit. With the help of crowdworkers we extract requirements from the unstructured free text submissions and comments. Then, we systematically analyze the applicability of five standard recommender algorithms for supporting such a narrative in recommender systems. We evaluate these approaches by comparing the human community suggestions for narrative requests with purely algorithmic recommendations. To optimize the approaches and improve the initial results, we refine the computed recommendations by applying post-filtering and re-ranking strategies.

As the initial results of this article suggest, traditional recommender algorithms exhibit great potential for improvement when presented with a narrative, as they lack the proper means to include a priori specified requirements in the recommendation process. However, we demonstrate that we can improve all recommender approaches by applying post-filtering and re-ranking strategies using metadata available in the narrative of the initial requests on reddit. The findings of this article and the publication of the reference dataset enable researchers to conduct independent analyses, advancing the state of research in the context of narrative-driven recommendations.

3.5.1 Abstract

Recommender systems have become omni-present tools that are used by a wide variety of users in everyday life tasks, such as finding products in Web stores or online movie streaming portals. However, in situations where users already have an idea of what they are looking for (e.g., *'The Lord of the Rings', but in space with a dark vibe*), most traditional recommender algorithms struggle to adequately address such a priori defined requirements. Therefore, users have built dedicated discussion boards to ask peers for suggestions, which ideally fulfill the stated requirements. In this paper, we set out to determine the utility of well-established recommender algorithms for calculating recommendations when provided with such a narrative. To that end, we first crowdsource a reference evaluation dataset from human movie suggestions. We use this dataset to evaluate the potential of five recommendation algorithms for incorporating such a narrative into their recommendations. Further, we make the dataset available for other researchers to advance the state of research in the field of narrative-driven recommendations. Finally, we use our evaluation dataset to improve not only our algorithmic recommendations, but also existing empirical recommendations of IMDb. Our findings suggest that the implemented recommender algorithms yield vastly different suggestions than humans when presented with the same a priori requirements. However, with carefully configured post-filtering techniques, we can outperform the baseline by up to 100%. This represents an important first step towards more refined algorithmic narrative-driven recommendations.

3.5.2 Introduction

The practical applications of recommender systems are manifold. In general, they are tools that help users to find and discover items of interest in large collections, such as books, movies, or people. In a common collaborative filtering scenario, a recommender system makes use of a user's history and predicts new items that user is likely to read, watch, or connect to.

Problem. Often, users already have vague to specific ideas about the desired entities they want to be recommended. More precisely, users often seek recommendations that fit arbitrary criteria, such as movies that evoke certain emotions or have a surprising ending, instead of obtaining suggestions purely based on their (and other users') histories of interactions within a given system. These criteria represent the **narrative** of a recommendation request. Recommendations generated by incorporating such a narrative are referred to as narrative-driven recommendations [Bogers and Koolen, 2017] and also build the foundation for conversation-based recommendation approaches used in chat- and voice-bots. Due to the lack of automated recommender systems that can accurately calculate such recommendations, users have built various discussion boards on the Web to ask peers for suggestions. For example, as of March 2017, there were 190,000 discussion threads with nearly 25,000 threads containing requests with a narrative for interesting books on the social cataloging website LibraryThing[20] [Bogers and Koolen, 2017]. Also, there are several subreddits on reddit.com, where users can ask for, for example, video game, movie, or board game suggestions. Requests for movie recommendations can look as follows: *"[...] Movies with the genre 'Crime' [...] like 'Nightcrawler' and 'Prisoners' [...] And it is great if there is any form of plot twists"*[21]. The (free-form) narrative of such requests defines several different elements, such as positively or negatively associated movies (i.e., *Nightcrawler*, *Prisoners*), preferred as well as unwanted genres (i.e., *Crime*), and specific keywords that define desired or undesired attributes/keywords of the movie (i.e., *plot twists*) [Bogers and Koolen, 2017].

Approach. In this paper, we systematically analyze the suitability of five standard recommender algorithms for supporting such a narrative in recommender systems. For our evaluation, we compare human suggestions for requests that provide a narrative with purely algorithmic recommendations.

To that end, we first compile an evaluation dataset by collecting and parsing narrative requirements from users of the subreddit r/MovieSuggestions[22].

[20] https://www.librarything.com
[21] https://www.reddit.com/r/MovieSuggestions/comments/3fvycr
[22] https://www.reddit.com/r/MovieSuggestions

We extract requirements from the unstructured text of submissions and comments with the help of crowdworkers and make our dataset available online[23] for future research. Next, we implement a recommender framework based on ratings, reviews and textual information of movies available on the Internet Movie Database[24] (IMDb). We calculate recommendations using the following five algorithms for our analysis: item-based collaborative filtering (CF), matrix factorization (MF), a content-based filtering approach based on TF–IDF similarities (TF–IDF), document-level embeddings (Doc2Vec), and a network-based approach (NW). In addition, we extract movie suggestions generated by IMDb, which we use as an empirical baseline (IMDb baseline). We apply post-filtering and re-ranking strategies using metadata from IMDb to refine the computed recommendations. Finally, we evaluate the five recommender approaches by measuring the overlap between their recommendations and the suggestions from users in our evaluation dataset from reddit. Our initial results suggest that traditional recommender algorithms exhibit great potential for improvement when presented with a narrative, as they lack the proper means to include a priori specified requirements in the recommendation process. Further, we demonstrate that we can improve all recommendation approaches (including existing empirical IMDb recommendations) by applying post-filtering and re-ranking strategies using metadata available in the narrative of the initial requests on reddit.

Contributions. With our analyses, we make the following contributions. First, we publish a reference dataset, which enables researchers to conduct independent analyses, advancing the state of research in the context of narrative-driven recommendations. Second, we evaluate the performance of five well-studied recommender approaches on our reddit evaluation dataset, containing a total of 1,480 recommendation requests that provide a narrative. Third, we demonstrate how to improve narrative-driven recommendations by introducing post-filtering and re-ranking techniques and analyze their importance for each of our five implemented recommendation approaches.

[23] https://www.rbz.io/datasets
[24] https://www.imdb.com

3.5.3 Related Work

Traditional Recommender Systems. There exists a vast variety of studies about recommender systems and algorithms (e.g., [Adomavicius and Tuzhilin, 2011; Adomavicius et al., 2011; Basu et al., 1998; Bogers and Koolen, 2017; Christakopoulou et al., 2016; Christakou et al., 2007; Ghosh et al., 1999; Hariri et al., 2013; Lamprecht et al., 2015; Mahmood and Ricci, 2009; Mak et al., 2003; McGinty and Reilly, 2011; Perny and Zucker, 2001; Seitlinger et al., 2015]). However, we still only have limited insights into the quality and suitability of traditional recommender algorithms for calculating narrative-driven recommendations. Typically, traditional research in recommender systems focuses on algorithmic advantages in common scenarios, such as applying users' histories and profiles to compute recommendations [Christakou et al., 2007; Ghosh et al., 1999; Mak et al., 2003; Perny and Zucker, 2001].

Context-Aware Recommender Systems. To compute recommendations that are well suited to the current needs of a user, context-aware recommender systems use contextual information, such as the time of the day or the current location or interests of the user, besides user profiles and histories [Hariri et al., 2013]. In a context-driven environment, Adomavicius et al. [2011] introduced REQUEST, which is a query language for customizing recommendations based on users' personalized recommendation needs. Hariri et al. [2013] proposed a query-driven context-aware recommender system that considers user profiles, item representations, and contextual information, such as interests or needs of a user in a specific situation.

A context-aware support vector machine for application in a context-dependent recommender system was proposed by Oku et al. [2006]. The authors found that for information recommendation it is important to consider the situations or conditions which influence the users' decisions (e.g., time of day, weather, physical condition).

In the study of Adomavicius et al. [2005], the authors presented a multidimensional recommendation model that is based on additional contextual information, such as profiles and aggregation hierarchies. They evaluated their approach on a movie recommender by exploiting contextual information, such as when a movie was seen, where, and with whom. They empirically demonstrated that this contextual information can improve the recommendations.

Basu et al. [1998] conducted a study on IMDb data, in which they proposed a recommender approach that exploits both user ratings and content information using collaborative, content, and hybrid features. Lamprecht et al. [2015] analyzed how IMDb recommendation networks support alternative information retrieval strategies, such as browsing. The authors showed that current recommendation networks are poorly navigable and require further improvements. This shows potential for providing context-aware recommender systems that involve the current needs of a user without the need of clicking through poorly navigable recommendation networks until finding a more or less fitting movie.

Adomavicius and Tuzhilin [2011] argued that relevant contextual information is important when providing recommendations. Such contextual information can be obtained explicitly (i.e., users provide additional information) or implicitly (i.e., system implies the context automatically from the given requirements). To that end, the authors introduced pre- and post-filtering techniques for capturing relevant context during the recommendation process. They used these methods for selecting a relevant set of data and for filtering out irrelevant recommendations or adjusting the ranking of the obtained recommendation list based on a given context. They discussed the notion of context and how it can be modeled, and conducted an empirical analysis using movie data regarding only the combination of several pre-filters. In this paper, we follow up on their ideas.

In contrast to the study of Panniello et al. [2009] that constitutes a first step towards the comparison of pre- and post-filtering using just one contextual variable for each applied dataset, we introduce and combine several post-filters and evaluate their utility in the context of narrative-driven movie recommendations.

Narrative-Driven Recommender Systems. Bogers and Koolen [2017] presented a specific context-aware recommendation scenario called narrative-driven recommendation. In such a scenario recommendations are computed based on past transactions of users, and a narrative description of the current needs and interests of users. Narrative-driven recommendations are related to conversational-based recommender systems, where users ask for suggestions in a community and other users then come up with suggestions and possible explanations for their choices [Christakopoulou et al., 2016; Mahmood and Ricci, 2009; McGinty and Reilly, 2011].

[Bogers, 2015] analyzed the movie discussion threads from the IMDb message boards that contain requests for movies to watch. The author found that content (e.g., movie description), different types of metadata (e.g., genre, language, release year), and searching for a movie by describing its content (e.g., in cases where users forgot the movie title) are important for movie selection practices.

In contrast to previous work, we present the first in-depth analysis and evaluation of recommender algorithms to support narratives for the computation of recommendations.

3.5.4 Reddit Narratives Evaluation Dataset

On r/MovieSuggestions, users ask other users for movie suggestions by describing, in natural language, what they are looking for. For example, typical posts include questions such as *"[...] Really dark, slow paced movies with minimal story, but incredible atmosphere, kinda like 'Drive' (2011), 'The Rover' (2014), or 'No Country for Old Men' (2007)? [...]"*[25]. The narrative of this example includes references to three "positively associated" movies (i.e., *Drive*, *The Rover*, *No Country for Old Men*) and several keywords that define the gist of the plot (i.e., *incredible atmosphere, dark, slow paced*, *minimal story*). As these requests are written in free-form text, the amount of information that can be leveraged for calculating recommendations varies. For example, users sometimes include detailed lists and descriptions of movies that they previously did (or did not) enjoy in their requests. Other times, only a single movie is referenced. Further, users frequently provide keywords in the narrative, which should apply to

[25] https://www.reddit.com/r/MovieSuggestions/comments/3kjrus

the suggestions (e.g., *"[...] Movies that will make me want to cry [...] like 'Extremely Loud and Incredibly Close' "*[26] with the keyword *cry* and one desired movie, or *"[...] Movies that take place primarily in one room or building. [...] Examples: Exam, Circle, Hateful Eight, Die Hard [...]"*[27] including the keywords *one room or building* and some desired movies). Other users then suggest appropriate movies by writing comments to the original post. Note that recommendations on r/MovieSuggestions are usually generated only considering the information provided in each submission, ignoring previous interactions or requests of users, limiting the amount of available information (see Table 3.17 for a more detailed characterization of our dataset).

Requests with a Narrative. To compile a dataset suitable for the evaluation of narrative-driven recommendations, we extracted all submissions from r/MovieSuggestions that (i) were posted between August 14, 2011 and August 1, 2017[28], (ii) had received at least ten comments, and (iii) had a score (i.e., the sum of up- and down-votes) greater than zero (3,640 of 23,484 submissions after filtering). Additionally, we extracted all comments to these submissions that had a score greater than zero, which we used as indicator for good recommendations (24,851 of 201,298 comments after filtering). For the compilation of the dataset, we asked crowdworkers to match the movies, genres, actors and other keywords mentioned in the reddit narratives to their corresponding entries on IMDb. The IMDb website provides a wide variety of information about movies and TV shows, such as genres, descriptions, trailers, plot summaries, as well as details about the cast, producers, and writers. In February 2017 the publicly available dataset[29] included information about 4.1 million titles and 7.7 million people.

Crowdsourcing Requests and Suggestions. To obtain a structured set of user requests and suggestions, we asked crowdworkers to annotate the unstructured text of the previously extracted submissions and comments from r/MovieSuggestions after filtering (see Table 3.17 for more details).

[26] https://www.reddit.com/r/MovieSuggestions/comments/11ycep
[27] https://www.reddit.com/r/MovieSuggestions/comments/4va9p8
[28] The dump is available at https://files.pushshift.io/reddit [Baumgartner, 2015]
[29] https://www.imdb.com/interfaces

To that end, we designed four micro tasks on the crowdsourcing platform CrowdFlower (now Figure Eight).[30] First, in the `SUBMISSIONS` task, we asked crowdworkers to identify all movie titles in each submission. Second, in the `SENTIMENT` task we asked crowdworkers to specify the sentiment of the user with respect to a movie mentioned in a submission (i.e., positive

[30] https://www.figure-eight.com

Table 3.17: **Reddit Evaluation Dataset Characteristics.** This table lists the statistics of our reference dataset, which we compiled using data from r/MovieSuggestions and crowdworkers on Crowdflower to extract structured data from the unstructured text of the submissions and comments.

#Submissions	1,480
Average Submission Score	11.78
#Movies in Submissions	5,521
#Unique Movies in Submissions	1,908
#Submissions with Desired Movies	1,480
#Submissions with Undesired Movies	75
#Keywords in Submissions	4,492
#Unique Keywords in Submissions	1,878
#Submissions with Desired Keywords	1,198
#Submissions with Undesired Keywords	153
#Genres in Submissions	762
#Unique Genres in Submissions	26
#Submissions with Desired Genres	491
#Submissions with Undesired Genres	61
#Actors in Submissions	100
#Unique Actors in Submissions	79
#Submissions with Desired Actors	75
#Submissions with Undesired Actors	6
#Comments	21,032
Average Comment Score	2.88
#Movie Suggestions in Comments	43,402
#Unique Movie Suggestions in Comments	6,071
Average #Movie Suggestions per Submission	29.33
Average #Movie Suggestions per Comment	2.48

or negative association to the requested suggestions). We defined *positively associated* movies as movies that users liked or where they stated that they were looking for movies similar to these. Analogously, we defined *negatively associated* movies as movies that users disliked or where they stated that they were not looking for similar movies. Third, in the `KEYWORDS` task we asked crowdworkers to identify additional information about the user's preferences in each submission's text (i.e., keywords). To extract these keywords, we provided the crowdworkers with a list of keyword types containing, for example, genres, movie settings, and events.[31] We asked the crowdworkers to identify *positively associated keywords* (i.e., keywords which should apply to the recommendations) and *negatively associated keywords* (i.e., keywords which should not apply to the recommendations). Finally, in the `COMMENTS` task, crowdworkers identified all movie titles in the comments to each submission.

A minimum of three separate crowdworkers worked on each submission in the `SUBMISSIONS` task. Where there was high disagreement among the workers, we requested judgements from two additional workers. Three workers worked on each movie in the `SENTIMENT` task and each comment in the `COMMENTS` task. In the `KEYWORDS` task, five distinct workers extracted keywords from each submission. We ensured the quality of the crowdworkers' output by requiring an entry-quiz for each task. Additionally, we continuously assessed workers via test questions.

Post-Processing. To obtain a well-curated dataset for the training and evaluation of narrative-driven recommendations, we carried out several manual and semi-automatic post-processing steps.

First, we manually reviewed all submissions from the `SUBMISSIONS` task and all comments from the `COMMENTS` task that did not have the crowdworkers' full agreement on movie titles. The crowdworkers fully agreed on the movie titles in 1,205 submissions and 16,893 comments, and they disagreed on titles in 457 submissions and 7,958 comments, which we then manually reviewed. During this step, we also removed submissions and comments without movie titles.

[31] The full list of keyword types included *genres, actors, movie directors, movie characters, movie producers, movie production companies, events or special occasions, movie settings,* and *other movie characteristics.*

Second, we aggregated the answers from the SENTIMENT and KEYWORDS tasks. In the SENTIMENT task we applied a majority vote whereas in the KEYWORDS task we first split the keyword strings provided by the workers into single keywords. Then, we retained all keywords identified by at least two out of the five workers.

Third, we automatically and unambiguously matched 1,298 movie titles from the SUBMISSIONS and 5,695 movie titles from the COMMENTS task to movie titles from IMDb. We then manually reviewed all movie titles that could not be automatically mapped to IMDb. In cases where more than one (or no) movie existed with the exact same movie title, we matched the movie using contextual information of the submission and the comments. In cases where we did not have sufficient information to unambiguously map movies, we removed them from our reference dataset.

Fourth, we automatically identified all common movie genres and actors in the keywords by matching them to the 25 genres and 294,533 actors available in our IMDb data.

Finally, we removed all movies from the submissions and comments that are not present in our IMDb data. Further, we removed submissions that did not contain any positively associated movie and that did not receive at least ten unique movie suggestions in the comments. After the last preprocessing step, our reference dataset[32] consists of 1,480 movie recommendation requests and 43,402 corresponding suggestions, as noted in Table 3.17.

3.5.5 Experimental Setup

Our recommendation framework (see Figure 3.13) (i) uses one or more movies as input data, (ii) implements five different recommender algorithms to compute a candidate set of recommendations, and (iii) applies several post-filtering and re-ranking strategies, based on metadata from IMDb to calculate a final list of (top ten) recommendations.

[32] Note that on our website https://www.rbz.io/datasets, we also provide necessary information about the mapping of genres and actors, and an extended version of our dataset without thresholds for the number of suggestions or the number of positively mentioned movies.

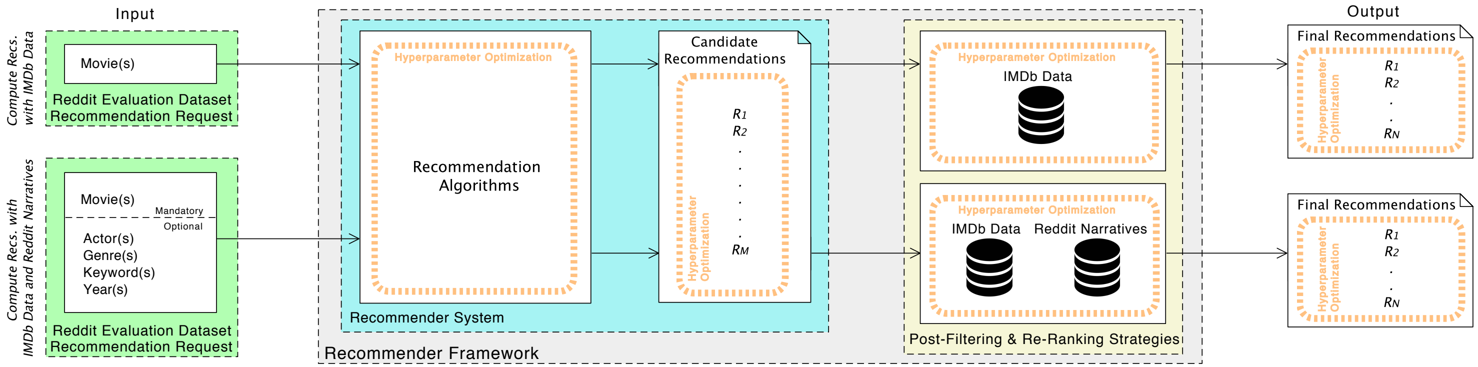


Figure 3.13: **Experimental Setup.** The recommender framework accepts several input parameters (see *Input*), extracted from the narrative of a recommendation request (e.g., reddit submissions). We distinguish between requests that only provide information about desired movies (see *Compute Recs. with IMDb Data*) and requests that include more detailed information from their narratives (see *Compute Recs. with IMDb Data and reddit Narratives*). The input parameters are then fed into the implemented recommender algorithms (see *Recommender System*), which calculate a first list of candidate recommendations. We then apply post-filters (see *Post-Filtering & Re-Ranking Strategies*) based on IMDb Metadata, or IMDb Metadata and reddit narratives, to provide a re-ranked list of recommendations (see *Output*), which better reflects the requirements defined in the narrative of the recommendation request. For all parts that are highlighted in orange (see *Hyperparameter Optimization*), we conduct an extensive grid search over relevant parameter configurations to find the optimal parameter settings.

To assess the importance of narratives for the calculation of recommendations we further calculate an alternative final recommendation list by applying the structured input (in the form of e.g., actors and keywords) from a given reddit narrative in the post-filtering and re-ranking step.

Finally, we evaluate both lists by comparing them to human suggestions from our reddit evaluation dataset (see Section 3.5.4).

Hyperparameter Optimization. To analyze if and to what extent traditional recommender approaches can support narratives we aim at making as few assumptions as possible and take a data-driven approach. Thus, we conduct an extensive cross-validation over various configurations of the parameters of the algorithms (see framework components highlighted in orange in Figure 3.13). Specifically, we optimize (i) hyperparameters for the algorithms, such as similarity measures or regularization parameters, (ii) the lengths of the initial and the final recommendation lists, and (iii) hyperparameters of the post-filtering and re-ranking mechanisms, such as overlap measures or functional forms for various scores. We discuss the optimal parameter configurations that we obtain along with introducing a given framework component.

IMDb Movies and Ratings. To implement the recommender algorithms we use data from IMDb. Note that training of recommender algorithms directly on our reddit evaluation dataset is not viable due to the sparsity of data. We leave this option open for future work when more data is available.

In addition to the publicly available IMDb dataset, we collect user reviews and individual ratings for all movies on IMDb. For our experiments, we only consider movies and discard all other types available on IMDb, such as TV series or single TV episodes. To minimize noise and to allow for fair comparisons between the different approaches, we only *keep movies* that have (i) more than 1,000 user ratings, (ii) at least one user review, (iii) a movie description, and (iv) at least one person in the cast. In contrast, we do not *remove users* with small numbers of ratings, as this preprocessing step does not improve our results. We obtain the rating thresholds for movies (1,000) and users (no limit) via grid search.[33] For more details see Table 3.18. Further, we compute centered ratings [Desrosiers and Karypis, 2011; Resnick et al., 1994] by removing user and item bias which improves the overall performance of all implemented recommender approaches.

Recommender Strategies

We generate recommendations by computing similarities between an input movie and all other movies available in our IMDb dataset. Each

[33] We perform the grid search over 0 to 10,000 movie ratings in increments of 500, as well as 0 to 500 user ratings with increments of 10.

Table 3.18: **IMDb Dataset Characteristics.** This table describes the features of the dataset that we used for computing the recommendations of our implemented recommender algorithms.

#Movies	11,578
#Ratings	144,021,151
Average #Ratings per Movie	$\approx$ 12,439.21
#Users with Ratings	1,144,136
Average #Ratings per User	$\approx$ 125.88
#Reviews	1,880,837
Average #Reviews per Movie	$\approx$ 162.45
#Users with Reviews	598,247
Average #Reviews per User	$\approx$ 3.14
#Credits	667,279
#People in Cast and Crew	322,881
#Actors	294,533
Average #Actors per Movie	$\approx$ 25.44
Average #Movies per Actor	$\approx$ 2.27
#Genres	32,767
#Unique Genres	25
Average #Genres per Movie	$\approx$ 2.83
#Plot Keywords	1,124,510
#Unique Plot Keywords	89,003
Average #Plot Keywords per Movie	$\approx$ 97.12

recommender algorithm determines how and with which data we calculate similarity. As similarity measures we use cosine similarity and an inverse of Euclidean distance and select the best performing measure via cross-validation. In cases where we have more than one input movie we aggregate similarity values. Hence, for each movie in our IMDb data, we add all similarities for all positively associated input movies. Our cross-validation yields better results when we do not subtract negative input movies for the aggregation of similarity values. We call the aggregated similarities *algorithmic score*. Thus, the output of each approach is a ranked list of candidate movies with their corresponding algorithmic scores. We conduct experiments with the following five approaches:

Item-Based Collaborative Filtering. This approach finds similar movies to the movies that a user liked [Sarwar et al., 2001]. Thus, we use the IMDb user-ratings vectors of two movies to compute their similarity [Sarwar et al., 2001]. The best performing similarity measure for this approach is cosine similarity.

Matrix Factorization. This approach is a well-established method that approximates a ratings matrix with the product of two matrices, one connecting users to factors representing their preferences, and another connecting movies to factors representing their properties [Koren, 2008; Paterek, 2007; Salakhutdinov and Mnih, 2007, 2008]. In this paper, we factorize the IMDb user-ratings matrix in a standard manner by minimizing a regularized squared error with a stochastic gradient descent [Funk, 2006]. We then use cosine similarity (determined via hyperparameter optimization) to compute similarity between the obtained movie factors.[34]

Content-Based Filtering with TF–IDF. We use this approach to find similar movies by calculating similarity between movies using their descriptions and user reviews [Adomavicius and Tuzhilin, 2005]. Hence, we compute the term frequency–inverse document frequency score [Salton and McGill, 1986] of terms in the description and user reviews for each movie. To compute the similarity between movies we use normalized

[34] We have tested different numbers of factors ranging from 100 to 1,000 in steps of 100, learning rates between 0 and 0.1 in steps of 0.001, and regularization parameters from 0 to 0.1 in steps of 0.01. We obtain the best results for MF with 500 factors, a learning rate of 0.002, and 0.02 as regularization parameter.

TF–IDF vectors and the reciprocal of Euclidean distance (determined via hyperparameter optimization). We receive the best results with unigrams and bigrams, no cut-off threshold for less frequent terms, and with a maximum of 500 features for the TF–IDF vectors.[35]

Document-Level Embeddings with doc2vec. Similar to the TF–IDF approach, we use movie descriptions and reviews as basis for this approach. `doc2vec` was first proposed by Le and Mikolov [2014] and is an enhancement of `word2vec` [Mikolov et al., 2013a], extending the learning of embeddings from words to documents. We use `doc2vec` to generate a document vector for each movie and use these vectors to compute similarities between movies. We obtain the best results with a feature vector dimensionality of 500 and cosine similarity.[36]

Network-Based Recommendations. We use this approach to find movies with similar casts and crews by creating a bipartite graph between movies and people involved in those movies. Specifically, we connect each movie to all cast and crew members including actors, cinematographers, composers, costume designers, directors, editors, producers, production designers, special effect companies, and writers. We calculate similarity between movies by counting common neighbors in the bipartite graph [Huang et al., 2005].

IMDb Baseline. We collect all movie suggestions on IMDb[37] for each movie in our dataset to determine if and to what extent existing (empirical) recommender systems are suitable to address a narrative. IMDb provides a maximum of twelve recommendations per movie. We use these recommendations for all (desired) input movies in the narrative of each submission. Note that IMDb does not provide any ranks or numerical values quantifying the quality of each recommendation.

[35] To obtain this configuration we conducted a grid search experiment over different n-grams [Brown et al., 1992] (i.e., $n = 1, 2, 3$), several cut-off values for terms with a low document frequency from 0 to 0.1 in increments of 0.001, and different numbers of TF–IDF features ranging from 0 to 1,000 in steps of 100.

[36] To obtain this configuration we conducted a grid search experiment with different similarity measures, and different feature vector sizes ranging from 0 to 1,000 in steps of 100.

[37] For an example see “More Like This” on `https://www.imdb.com/title/tt0076759`

Post-Filtering and Re-Ranking

We further refine the algorithmic recommendations by defining several post-filtering approaches, which allow us to include (i) additional metadata from IMDb, and (ii) optionally reddit narratives in our recommendations. Again, for evaluation of various post-filters we pursue a data-driven approach and conduct extensive cross-validations over multiple configurations. This allows us to evaluate the importance of the individual post-filters as well as the interactions between different post-filters.

Specifically, with our post-filtering techniques we modify the calculated recommendation list by (i) removing irrelevant recommendations for a given movie, and (ii) re-ranking the obtained list. In general, the more properties (e.g., genres, keywords, actors) the candidate movies have in common with a given input movie, the higher they get ranked. For example, we compute the overlap of genres of all input movies and a candidate movie. With all scores calculated we re-rank the candidate lists by combining algorithmic scores of each candidate recommendation with the corresponding post-filtering scores to compile a final recommendation list. We evaluate the resulting (final) list by comparing it to human suggestions from our reddit evaluation dataset. When limiting our final recommendation list to a total of ten movies to be displayed, we achieve the best results with 500 candidate recommendations.[38]

Post-Filtering and Re-Ranking with IMDb Data. With IMDb metadata we re-rank candidate recommendations with the following scores:

IMDb Popularity and Rating Score. Following the intuition that users are generally more interested in higher and more frequently rated movies, we introduce this score which combines the average IMDb rating (*rating score*) of a candidate movie and the number of ratings received on IMDb (*popularity score*). We experiment with various functional forms for the computation of both the average rating and the number of ratings. Specifically, we calculate logarithmic, square root, quadratic, and cubic scaling

[38] We determine the length for the candidate list with a grid search over the range from 100 to 1,000 movies in steps of 100.

and achieve the best results with the following functional form: $\log_2(R_i)\overline{r}_i$, where $\overline{r}_i$ is the average rating, and R_i is the number of ratings of movie i.

IMDb Genre Score. Here, we follow the intuition that users prefer movies of similar genres to the specified movies and calculate the IMDb genre score for each candidate movie. As part of our hyperparameter optimization, we compare several overlap measures, including Jaccard's coefficient, cosine similarity, Sørensen-Dice coefficient, and simple matching coefficient. We achieve the best results with similar scaling and normalizing of the overlap between the genres of the candidate movie and individual positively associated movies from the request so that $S_{\text{iGenre}}(i) = \sum_{j \in I_{\text{pMovie}}}(|G_i \cap G_j|^2/(|G_i||G_j|))$, where I_{pMovie} is the set of positively associated input movies. The inclusion of negatively associated input movies does not improve our results.

IMDb Year Score. We assume that users want to watch movies from similar time periods unless explicitly stated otherwise. Thus, we introduce the IMDb year score, where candidate movies released closer in time to the input movies receive higher scores. We set this score to 1 for a candidate movie with the smallest difference in release year to one of the input movies. We then linearly scale the year score until we reach 0 for a given maximal difference in release years. We obtain the best results with a release year normalization of 50 years.[39]

IMDb Keyword Score. For our recommender framework, keywords are words or phrases that represent a very specific attribute of a movie. For the IMDb keyword score, we use the plot keywords from IMDb and compute the overlap of all plot keywords of a candidate movie and the plot keywords of the input movies. Following a grid search approach we determine Jaccard's coefficient as the most suitable overlap measure while ignoring plot keywords of negative input movies.

IMDb Predecessor and Successor Filters. We assume that users do not want to receive a list of predecessors or successors of the specified input movies as they are likely familiar with the whole series. Hence, we remove predecessor and successor movies from our recommendation lists. For

[39] Identified via grid search over the year-range from 20 to 100 in steps of 10.

example, if users ask for movies similar to *The Hunger Games: Catching Fire* we remove *The Hunger Games* and *The Hunger Games: Mockingjay - Part 1 & 2* from our recommendation list.

Combining Scores. To compute the final score for each candidate movie we first normalize all computed scores by their highest values, so that (for each score individually) the movie with the highest score receives the value 1. Second, as post-filters are not equally important across our approaches, we multiply the scores with weights, reflecting their influence for the re-ranking of the recommendation lists. We conduct a grid search experiment over all combinations of weights between 0.0 and 1.0 in steps of 0.2, and select the setup that yields the best results in our experiments. Finally, we sum up all weighted scores to obtain the final score for each movie.

Post-Filtering and Re-Ranking with Reddit Narratives. For the final step of our evaluation we incorporate metadata, available in the narrative of the initial reddit submission, into our recommendations using additional post-filters. Specifically, we use keywords, genres, actors, and years given in the narrative of the movie suggestion requests in our reddit evaluation dataset. Note that we can calculate post-filtering scores from reddit narratives only if users explicitly provided positively/negatively associated attributes or keywords (e.g., actors or genres) in a recommendation request (see Table 3.17). With all scores calculated we re-rank the candidate lists (500 candidates) again by combining all IMDb post-filtering scores of each candidate recommendation with the corresponding narrative-based post-filtering scores to compile a final recommendation list (ten recommendations). Again, we evaluate the resulting (final) list by comparing it to human suggestions from our reddit evaluation dataset. To that end, we define and compute the following narrative-based post-filtering scores and evaluate their importance by conducting a grid search experiment:

Narrative-Based Genre Score. If genres are stated in the narrative of a request, we use them to calculate the narrative-based genre score for each candidate movie. We ran the same grid search experiment as we did for the *IMDb Genre Score* and determined that the same overlap metric yields the best results. In contrast to the *IMDb Genre Score*, we remove movies with undesired genres from our recommendation list.

Narrative-Based Year Filter. If users explicitly state year thresholds, we re-rank the recommendation list so that movies outside this range are moved to the end of the list.

Narrative-Based Keyword Score. We exploit keywords in a specific request (e.g., "surprising plot twist") to introduce the narrative-based keyword score. With this score we measure how well the description and the user reviews of a candidate movie reflect the keywords stated in a narrative. We find that counting the incidences of explicitly stated keywords in the description and all user reviews of the respective candidate movies yields the best results by conducting a grid search experiment. We aggregate the incidences for positive input keywords and subtract them for negative ones. Finally, we compute the narrative-based keyword score by normalizing over the number of words in the used texts.

Narrative-Based Actor Filter. To reflect the requirement of only recommending movies with specific actors, we introduce the actor filter. We re-rank the list of movie recommendations by counting how many of the positively stated actors appear in the respective movies. Further, we remove all movies with actors that users explicitly specified as undesired.

Combining Scores. To combine all narrative-based post-filtering scores we use the same method as for the IMDb post-filtering scores.

Evaluation

We evaluate the implemented approaches on our reddit evaluation dataset. Specifically, we use the narrative from each submission to calculate movie recommendations and count the overlap between the movie suggestions of the reddit community, extracted from the replies to the corresponding submission (see Section 3.5.4) and our algorithmic movie recommendations. We calculate precision, recall, F1 score, normalized discounted cumulative gain (nDCG), and mean average precision (MAP) [Powers, 2011; Yilmaz et al., 2008b]. First, we chronologically split our reddit evaluation dataset into a validation (80%) and a test (20%) set (see Table 3.19). Second, we train our approaches on the IMDb data and use the reddit data from the validation set to conduct all grid search experiments for optimizing hyperparameters for the recommender framework. Finally, we evaluate the performance of the implemented approaches on the test set. We limit our final recommendation lists to ten movies.[40] To allow for a fair comparison we also limit the number of recommendations for our IMDb baseline to ten movies (picked at random, as recommendations are not ranked). First, we evaluate the standard algorithms with post-filters and scores calculated by using IMDb metadata. Second, we measure the performance improvements with the narrative-based post-filters and scores.

3.5.6 Results and Discussion

Post-Filtering and Re-Ranking with IMDb Data

Figure 3.14 depicts the results of the evaluation of our implemented algorithms for calculating recommendations for a given narrative using our reddit evaluation dataset. The transparent bars represent the means of the evaluation metrics over all submissions in the test set for a given approach using only IMDb-based post-filters, with the error bars showing the standard error. All of our analyzed approaches, while only relying on IMDb-based post-filters, manage to outperform the IMDb baseline (cf. horizontal dashed line in Figure 3.14). Doc2Vec performs best in all evaluation metrics with an F1 score of 0.117, which is more than twice

[40]This means that recall@10 and F1 score@10 have a mean upper limit of 0.34 and 0.51 respectively, as the average number of movie suggestions from the community per submission is 29.22 in the test set.

Table 3.19: **Evaluation Protocol.** Basic statistics of the validation set and the test set.

	#Submissions	Timeframe
Validation Set	1,184	08-2011 – 11-2016
Test Set	296	11-2016 – 07-2017
Overall	1,480	08-2011 – 07-2017

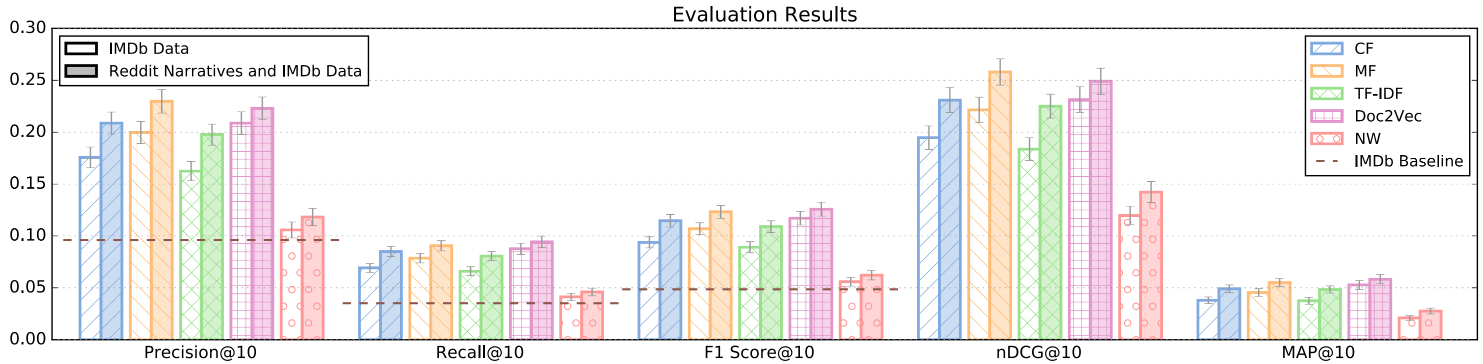


Figure 3.14: **Results.** This figure depicts the results of our evaluation, comparing our recommendations to the ones of the reddit community in our reddit evaluation dataset. We list the different evaluation metrics on the x-axis, with the corresponding evaluation metric values on the y-axis. The performances of the recommender algorithms with IMDb post-filters are represented by the transparent bars, while the filled bars depict the results for the approaches with additional narrative-based post-filters using reddit data. The grey error bars show the standard deviation of the evaluation metric over all submissions in the test set. All of our approaches outperform the IMDb baseline (dashed horizontal line). We can further improve the results by adding narrative-based post-filters, where Doc2Vec outperforms all other approaches with F1 scores more than twice as good as the IMDb baseline.

as good as the IMDb baseline, followed by MF with 0.107, CF with 0.094 and TF–IDF with 0.089, and NW, which performs consistently worst with an F1 score of 0.056, while still outperforming the IMDb baseline.

One possible reason for the moderate performance of NW might be that this approach is fundamentally based on the assumption that users want to see other movies with a similar cast. This inherent restriction appears to impair our results when incorporating the narratives provided by users. However, more research is warranted to further investigate this hypothesis, which we leave open for addressing in future work. MF and CF perform roughly twice as good as NW, possibly due to the larger amount of considered data. They are both based on user ratings and follow similar intuitions (i.e., both approaches favor frequently and highly rated movies), which could explain the similarity in the obtained results. TF–IDF, which is based on the text of movie descriptions and user reviews, performs similar to CF. Doc2Vec performs best of all approaches using the same data, which we attribute to the underlying mechanisms of the approach. Compared to TF–IDF vectors, word embeddings better incorporate latent factors in textual representations, leading to better similarity calculations and, therefore, better recommendations.

Importance of Post-Filters. We present the best-performing IMDb-based post-filter configuration for each approach by depicting the normalized score weight for each post-filter in Figure 3.15 (obtained by cross-validation), where a higher score signals higher importance of a given post-filter.

In case of CF, we obtain the best-performing configuration with a weight of 0.8 for the algorithmic score, a weight of 0.0 for IMDb popularity and rating influence, and relatively low weights of 0.4 for IMDb genre, keyword and year scores. For MF a higher algorithmic score weight (1.0) and high popularity and rating influence of 0.8 work best, while the year score is completely neglected and the IMDb genre and keyword scores are set to 0.2 and 0.4, respectively. The content-based approaches (TF–IDF and Doc2Vec) exhibit similar best-performing configurations with a 1.0 weight for the algorithmic scores, a 0.6 weight for the IMDb popularity and rating scores and a 0.2 weight for the IMDb year scores. The weights for the

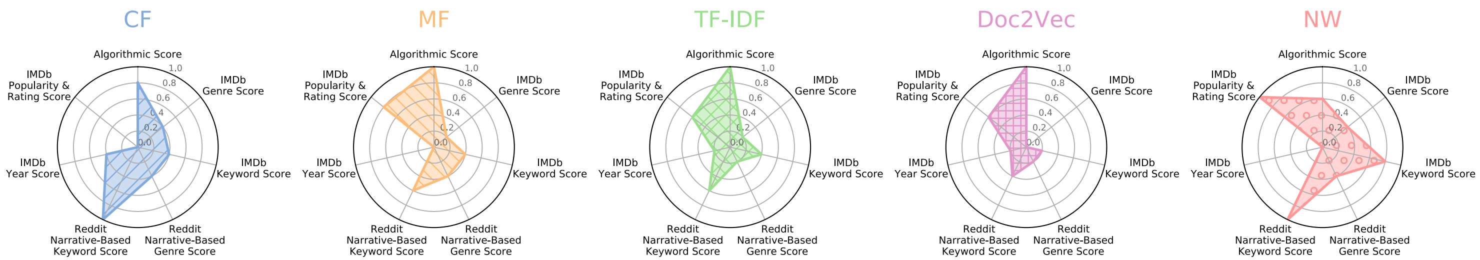


Figure 3.15: **Score Weights.** Each figure visualizes the score weight configuration of one approach. The algorithmic score and the IMDb popularity and rating scores are important characteristics across most of our approaches. Using narrative-based post-filters, the most important property are the keywords with weights up to 1.0. This also indicates that keywords are important for calculating narrative-driven recommendations.

IMDb genre and keyword scores range between 0.0 to 0.4. In contrast, NW mainly relies on keywords and popularity and rating influence with weights of 0.6 for the algorithmic score, 1.0 for the IMDb popularity and rating score and 0.8 for the IMDb keyword score. Similar to most other approaches, the influence of IMDb genres and years is quite low.

Findings. Our results reveal that for narrative-driven recommendation scenarios traditional recommender algorithms exhibit only minimal overlaps with human suggestions. Specifically, the algorithmic recommendations using post-filtering with IMDb metadata are computed by calculating similarities between the input movies and the movies from our dataset, while the narrative from reddit is neglected. However, additional information provided by users within their submissions appears to be crucial for the selection of appropriate movie suggestions. Users on reddit parse and consider this information, discerning their recommendations from algorithmic ones.

Post-Filtering and Re-Ranking with Reddit Narratives

In Figure 3.14 we also show the results of our experiments with post-filtering and re-ranking of the recommendations using the information from reddit narratives. Due to the fact that we now include narratives we can observe substantial improvements of our results when adding—and carefully configuring—post-filtering techniques (cf. transparent versus color-filled bars in Figure 3.14). Although not exhausting the potential for improvement, we raise F1 scores of our approaches to be more than twice as high as the IMDb baseline, except for NW. Again, we achieve the best results using Doc2Vec with an F1 score of 0.126, closely followed by MF with 0.123, CF with 0.115 and TF–IDF with 0.109.

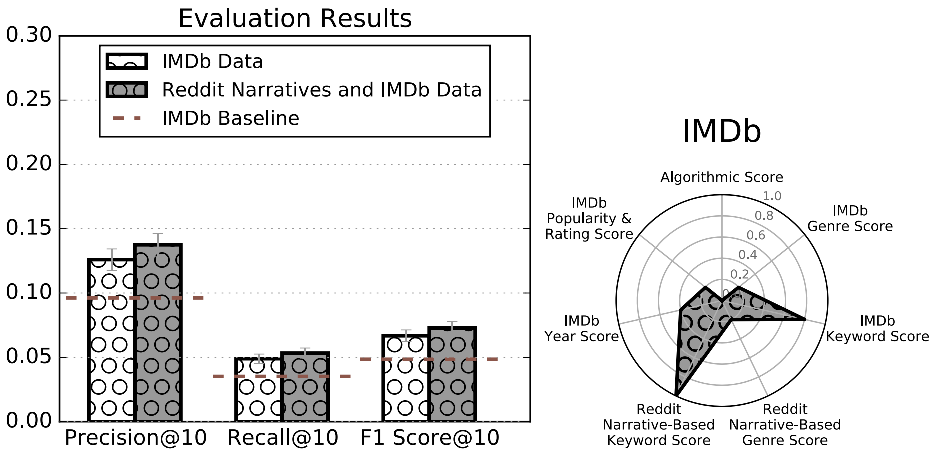
Importance of Post-Filters. Although the inclusion of the narrative information improves the recommendations, this additional information needs to be properly configured and strongly depends on the underlying algorithm. For all approaches, the best-performing configuration exhibits higher score weights for keywords extracted from the reddit narratives than for genres. For CF and NW, the narrative-based keyword score is very important, with configuration weights of 1.0, while it is 0.6 for MF and TF–IDF and 0.2 for Doc2Vec. For the narrative-based genre score CF, MF and NW have the same weights of 0.4, while the content-based approaches (TF–IDF and Doc2Vec) exhibit lower score weights of 0.2.

Findings. We find that carefully weighing the different post-filters, particularly in combination with the algorithmic, popularity and rating score, is important to maximize the benefit of the additional information contained in a given narrative.

Further, we find that for all approaches the most important narrative-based post-filter is the keyword score. From this result, we conclude that narrative recommendation requirements, provided in the form of keywords (i.e., the gist of a given text, such as short aspects of the story of a movie), are integral for achieving the best recommendations in our setup. We hypothesize that these keywords provide our post-filters with important information, that specifically helps to filter noise (i.e., unwanted movies) and steer our results towards more fitting movies. However, more research is warranted not only to confirm our hypothesis, but also to determine if additional post-filter or re-ranking strategies exist, for example, based on analyzing characteristics of recommendation requests, which could help to further improve our results.

Besides the narrative-based keyword score, the algorithmic and popularity and rating scores are also important for most of our approaches. This finding also strengthens our intuition that the configuration of algorithmic scores and post-filters is important for the computation of narrative-driven recommendations, and that it is not sufficient to simply apply filters on a given pool of existing recommendations as valuable information is lost and neglected in that process.

Except for NW, the influence of the IMDb genre and keyword scores are similarly low across all approaches. The least important score is the IMDb year score with weights ranging from 0.0 to 0.4. In fact, after manually inspecting our dataset, it appears that movies suggested by humans are more frequently from different years (even decades) than the movies mentioned in the recommendation requests (i.e., reddit submissions).



(a) Results (b) Score Weights

Figure 3.16: **Empirical Recommendations.** Figure 3.16a shows the results of our evaluation, comparing the empirical IMDb recommendations to the ones of the reddit community in our reddit evaluation dataset with IMDb post-filters (transparent bars) and with additional narrative-based post-filters using reddit data (colored bars). We list the different evaluation metrics on the x-axis, with the corresponding values on the y-axis, again. Figure 3.16b visualizes the best-performing score weight configuration of this experiment.

Applying Post-Filters on Empirical Recommendations

In addition to the datasets presented in this paper, we conduct another experiment to see if our post-filtering strategies can also improve our baseline IMDb recommendations. To that end, we apply all our post-filters on the IMDb baseline. We deploy the same evaluation setup as for our other previous experiments. First, we conduct a grid-search experiment to achieve the best-performing post-filter weights combination. Second, we apply all IMDb post-filters on the IMDb recommendations list and use the top ten recommendations for evaluation. The results, represented by the transparent bars in Figure 3.16a, reveal that additional IMDb metadata can be used to improve the resulting recommendations. Finally, we add post-filters with metadata from reddit narratives to the IMDb recommendations and further improve our results (see filled bars in Figure 3.16a), showing that it is possible to refine and improve recommendation algorithms to better support a given narrative using the post-filters presented in this paper.

The most important post-filters for this approach are the keyword scores from the IMDb data as well as from the reddit narratives (see Figure 3.16b). This further strengthens our finding that keywords provided in narratives are an important factor when re-ranking recommendations. Note that we do not have an algorithmic score for this approach as IMDb does not provide a ranking for their recommendations.

3.5.7 Conclusions and Future Work

In this paper, we analyzed and evaluated the potential of a selection of five (MF, CF, TF–IDF, Doc2Vec, NW) recommender algorithms as well as one empirical recommender approach (IMDb) to calculate narrative-driven recommendations. To be able to conduct our analyses, we crowdsourced a dataset from reddit for evaluating narrative-driven recommendations and made this dataset available to other researchers. Moreover, we re-ranked the computed recommendation lists via post-filtering techniques based on specific user requirements from the reference dataset. With our experiments we showed that (i) all implemented recommender approaches struggle to match human-based recommendations and that (ii) the incorporation of the information contained in the narratives (e.g., in the form of post-filters) can substantially improve the performance of recommender algorithms. However, we also showed that our post-filters have to be carefully configured to maximize the benefits of the added information, as the algorithmic score is an important feature across all approaches. Particularly, when applying post-filters on empirical data, we demonstrate that our post-filtering techniques can improve existing approaches, albeit limited due to the lack of an algorithmic score.

The post-filtering techniques applied in this paper are a first step into incorporating additional information provided by users into the recommendation process. For future work, we plan to investigate other similar heuristics for comparison with the ones used in this paper and to possibly obtain a further performance improvement. Moreover, we intend to extend existing algorithms by incorporating data from reddit narratives in the training phase in the form of, for instance, additional regularization terms. This could gain insight into how fast the recommendations adjust to the given recommendation needs of a user. Currently, the recommender algorithms can not be directly trained on the reddit data due to its sparsity but, as our results show, narrative information and the previous human suggestions represent a valuable information that should be leveraged already in the training phase.

Further, we plan on applying our methods to different domains, such as books, board games, or video games, to investigate whether different communities exhibit similar or different recommendation behaviors. Moreover, we will conduct a qualitative evaluation of our recommender framework to study if our suggestions are perceived as useful by the recommendation requesters. We are also dedicated to analyze additional post-filters, informed by characteristics of our reddit evaluation dataset, as well as expanding the arsenal of implemented recommender approaches, such as deep learning and different embedding approaches for the calculation of narrative-driven recommendations. Additionally, we aim on conducting experiments on reddit, by implementing a recommender bot that users can query for recommendations, while providing a narrative. Using this bot, we will be able to evaluate the importance of additional metrics, such as diversity, serendipity or novelty in the context of narrative-driven recommendations.

In this paper we present and publish a reference evaluation dataset, as well as a first analysis of post-filtering and re-ranking strategies for incorporating narratives into recommendations. We strongly believe that our reference evaluation dataset, as well as the presented experiments in this paper will help researchers and practitioners to develop new and improve existing recommendation approaches to better tackle the problem of narrative-driven recommendations, which also represents a fundamental problem in need of novel solutions for the advance of chat and voice bots.

4 Conclusions

Recommender systems have become omni-present tools that are used by a wide variety of users in everyday life tasks. Their practical applications are manifold, for example, finding products in Web stores, or movies on online streaming portals. In a typical recommendation scenario, a recommender system makes use of, for example, user profiles or transaction histories to generate suitable item suggestions. However, research about the usefulness of social information from multiplex networks as well as narrative aspects for the task of user to user or item to user recommendations was rare. Studies in this thesis broadened our understanding of signals available in social networks for the task of recommending interactions in dynamic online marketplaces. I presented the first analysis about multiplex networks and individual features from both an individual as well as a collective perspective to support trading interaction predictions in online marketplaces. Further, I evaluated the time dependencies of features to determine what types of features are the most useful ones at a certain point of time. The second part of this thesis concerns the usefulness of narrative features for movie recommendations with the ultimate objective of automating the process of narrative-driven recommendations. I compiled, published, and empirically analyzed a reference dataset based on movie suggestion requests from reddit. With the in-depth investigation of the dataset, this thesis provides insights into user preferences in narratives, how users tend to illustrate their needs, and whether positively associated aspects are more important for calculating recommendations than negatively associated aspects. Further, studies contained in this thesis present an evaluation of state-of-the-art recommender algorithms, including embedding approaches, to support narrative features for the computation of accurate movie recommendations. To refine the obtained results the framework presented in this thesis implements post-filtering and re-ranking strategies.

In the remainder of this chapter, I summarize the results and contributions of this thesis in Section 4.1, I provide an overview of the implications of this thesis in Section 4.2, followed by the limitations in Section 4.3, and, finally, I discuss potential new avenues for future work in Section 4.4.

4.1 Results and Contributions

In this section, I provide answers to the research questions defined in Section 1.3.

RQ 1: How can we utilize social network features in recommender systems?

Since our research community still have lacked insights about the usefulness of social signals originating from different network sources for the task of recommending sellers to buyers in an online marketplace, I raised and tackled this research question. To answer this question, I presented studies [Eberhard and Trattner, 2016; Eberhard et al., 2019a] that aim at predicting trading interactions between sellers and buyers in an online marketplace from four different perspectives, an online social, a location-based social, and a trading network, as well as different combinations of them. To that end, I incorporated and analyzed the structure of multiplex networks and the individual features from both individual as well as combined perspective. This allowed me to identify the impact of social features from the different networks from detailed to aggregated level. Further, I determined the robustness of features over time. For evaluation I compared state-of-the-art supervised and unsupervised link prediction methods. I found that the overall best trading prediction results could be achieved by combining all network sources. Online and location-based social network information on their own or in combination could achieve performance at an acceptable scale, which is convenient in specific settings (e.g., cold-start prediction). As expected, the trading network information improved the prediction performance strongly. Looking at the different applied methods, the supervised approaches were able to score quite well, especially utilizing an ensemble classifier. Also, the unsupervised strategies, which are more suitable in certain application settings where no ground truth information is available, scored sufficiently. Further, the results showed that topological features are more suitable than homophilic features for the given recommendation task. Regarding the time component of the features, I found that most of the used features outperform the baseline at each point in time.

RQ 2: How can we utilize narrative features in recommender systems?

First of all, I presented an in-depth empirical analysis to learn more about narrative recommendation requests and corresponding community suggestions from reddit [Eberhard et al., 2020]. To quantify the difficulty of the narrative-driven recommendation problem I analyzed the diversity of requests and their corresponding suggestions and evaluated the effects of positive vs. negative aspects. I found that users on reddit mainly focus their narrative requests on positive aspects, such as movies they liked or keywords that describe the movies they would like to see. The results of the presented study exhibited a significant correlation between positive movies and suggestions which indicates that users frequently describe similar requests with differing movie examples. Further, I found that community suggestions are oftentimes more diverse than requests, meaning that highly similar requests are frequently answered with highly diverse movie suggestions by the reddit community. These findings indicate that implementing automatic narrative-driven movie recommendations is a hard problem. Additionally, in studies [Eberhard et al., 2019b, 2020] presented in this thesis, I implemented a narrative-driven movie recommender framework to calculate recommendations for narrative requests on reddit using several state-of-the-art recommender algorithms, including embedding techniques. To refine and optimize the computed recommendations I applied post-filtering and re-ranking strategies using metadata from IMDb. To evaluate the implemented approaches I measured the overlap between the human suggestions from the reddit community and the purely algorithmic recommendations. To eventually answer this research question, I evaluated the importance of features extracted from narrative requests. I found that traditional recommender algorithms initially have exhibited great potential for improvement when presented with a narrative, as they have lacked the proper means to include a priori specified requirements in the recommendation process. Moreover, I showed that all implemented recommender approaches could be improved by applying post-filtering and re-ranking strategies using metadata available in the narrative of the initial requests. Finally, I found that positive movies and keywords have the strongest, whereas negative movie features the weakest predictive power for narrative-driven movie recommendations on reddit.

4.2 Implications

To support owners of online marketplaces in creating accurate product recommendations for their users, a better understanding of which social aspects are suitable predictors is essential. The gained insights into the topic of this thesis are also relevant for scientists who study recommender systems utilizing multiplex social networks. The empirical analysis about narrative-driven movie recommendations that I presented in this thesis is a first stepping stone for scientists and practitioners towards a better understanding about which aspects in narratives are most important and essential to compute accurate suggestions. I believe that the results of this thesis will benefit future research and provide actionable insights that serve as a valuable basis for further research in this area.

A Method to Combine Multiplex Networks for Link Prediction. Extending the well-studied research field of link prediction by incorporating social information from multiplex networks as presented in Eberhard and Trattner [2016] and Eberhard et al. [2019a] enables scientists to follow up and study effects arising due to this thesis. Additionally, the presented social feature analysis supports online marketplace owners to implement or improve a product or store recommender system by taking into account not only own trading data but also social aspects from external networks. Especially for cold-start scenarios in case, for example, when no trading data is available due to the novelty of a marketplace, a recommender system could benefit from multiplex external social network information.

Publicly Available Reference Dataset. To facilitate the process of analyzing and evaluating narrative-driven movie recommendations, a pre-processing step to extract entities, such as movie titles, genres, or keywords, from the narratives was essential. The crowdsourcing experiment presented in Eberhard et al. [2019b] helped to extract such entities and to bring the reddit narratives and comments into a structured form. Further, it helped to avoid the problem of automatic entity extraction for the moment. This compiled crowdsourced reddit dataset built the basis for the subsequent empirical analysis and the evaluation and optimization of the state-of-the-art recommender algorithms. Making this reference dataset publicly available enables researchers to conduct independent analyses, advancing the state of research in the context of narrative-driven recommendations.

Empirical Insights. The first in-depth empirical analysis of narrative-driven movie recommendations, as shown in Eberhard et al. [2020], gained insights into the difficulty of this specific context-aware recommendation scenario. The generated understanding of the importance of narrative aspects to compute accurate recommendations enables scientists to follow up on this thesis and extend the research in this direction. Additionally, administrators of streaming providers or online movie platforms derive benefit from the findings of this thesis. For example, they may adapt their recommender systems to support narratives—or at least narrative aspects in the form of keywords—to broaden the scope of requirements the system is able to cover and provide their customers with more accurate recommendations.

Narrative-Driven Movie Recommendation Guidelines. The empirical analysis about narrative-driven movie recommendations presented in this thesis has led to a number of hypotheses about narrative aspects for automatically computing accurate suggestions. Results of the conducted recommender experiments, as shown in Eberhard et al. [2019b] and Eberhard et al. [2020], proved most of the stated hypotheses. First, the experimental results emphasize that the problem of automatically computing accurate recommendations based on narratives is a difficult one.

Second, the results suggest to extract and focus on positive movie examples and positive keywords from narratives when computing algorithmic recommendations, since these features have the strongest predictive power. I strongly believe that the novel insights into narratives for recommender systems represent an important stepping stone towards novel applications, such as interactive recommender systems.

4.3 Limitations

In this section, I discuss the limitations the analyses in this thesis come with.

Availability of Datasets. For the social feature analyses in this thesis, a detailed social data basis from multiplex sources was required. Further, to challenge the task of predicting trading interactions between sellers and buyers, purchase logs of marketplaces were essential. Although such data is not principally publicly unavailable, the combination and specifically the conjunction of users between multiple network sources is rarely available. If such data is available, the findings in this thesis show how to reproduce the analysis to gain further insights and generalization of the results. All the narrative-driven recommender analyses in thesis are only based on movie suggestion requests from the online discussion board reddit. Consequently, it is not clear whether or not the obtained results hold true for other online movie discussion boards. Since the automatic extraction of important narrative aspects is part of future work, the help of crowdworkers was required to compile the dataset presented in this thesis. Without tools to automate this process, further datasets would entail considerable costs for crowdsourcing tasks.

Real-World Data. Since using data from Second Life, the trading interaction prediction experiments in this thesis are based on a virtual world. A rather important and relevant research direction concerns the understanding and modeling of digital and physical network structures and the behavior of actors therein. Consequently, it is unclear whether the findings about trading interaction predictions also hold true for data from the real world.

Generality of Empirical Findings. The empirical analysis as well as the evaluation and optimization of the state-of-the-art recommender approaches are based on a crowdsourced dataset from reddit. Hence, all findings of this analysis are limited to movie recommendation requests and suggestions on reddit. Further crowdsourcing experiments with different crowdworkers—independent from the ones used in this thesis—would strengthen the results and findings. To generalize the obtained findings about narrative-driven movie recommendations, experiments with data from further movie discussion boards are required.

Generality of Applications. This thesis concerns the link prediction problem of trading interactions utilizing social features and the problem of narrative-driven movie recommendations. To strengthen and generalize the obtained findings, besides the evaluation on similar datasets, investigation of the presented methods on other types of datasets is required. For example, the prediction of other interaction or relation types between users through social signals from multiplex networks, or the analysis of other narrative-driven recommendations, such as books, board games, or video games.

4.4 Future Work

To conclude this thesis, in this last section, I discuss potential avenues for future work to follow up on this thesis.

Expand Experimental Setup. For the analysis of trading interaction prediction in this thesis, I based my assumptions mostly on the link prediction and recommender systems research literature and common sociological factors indicating interactions. An interesting extension of this thesis would be to study the problem from a more theoretical and economical background. Building more grounded theoretical models would potentially also help to better understand the nature of this particular problem. For optimizing the implemented recommender approaches to support narrative requests, the post-filtering and embedding techniques applied in this thesis are a first step into incorporating additional information provided by users into the recommendation process. The analysis of other similar heuristics for comparison with the ones used in this thesis and to possibly obtain a further performance improvement would be an interesting path for future work. Such heuristics could be, for example, further embedding techniques, such as `item2vec` or `community2vec`.

Qualitative Evaluation. Due to the hardness of the narrative-driven recommendation problem, as discussed in this thesis, the evaluation results of the approaches revealed F1 scores that have some room for improvement. However, the ultimate goal of the recommender experiments was to see to what extend the community suggestions on reddit can be imitated. Due to this experimental setting there is no evidence about the quality of the algorithmic recommendations. There is still the open question whether or not the computed movie recommendations would be appreciated by the community. To that end, a qualitative evaluation of the introduced recommender framework would be a highly interesting direction for future research to study whether the suggestions are perceived as useful by the recommendation requesters. Depending on the results of such a qualitative evaluation, the performances of the approaches with F1 scores in the need of improvement in this thesis could be relativized.

Extend Research to Further Domains. To further validate the assumption that social signals from multiplex network sources help to improve link predictions, further studies on other datasets with different targets, for example, partner relationship prediction or any type of community creation, should be conducted. For the narrative-driven recommendations part of this thesis, analyzing and evaluating other domains, such as suggestion boards for books, board games, or video games, would be an interesting direction for future work. For example, the comparison of the characteristics between different communities and the investigation of which aspects are important in which community to provide relevant recommendations, would help to further broaden the understanding of the narrative-driven recommendation problem.

Automatic Entity Extraction. Another important path for future research is to supplement existing approaches by tackling the hard problem of automatically extracting important information from narrative requests by, for example, using natural language processing and named entity recognition. This would enable to easily compile further datasets about narrative-driven recommendations without the demand of expensive crowdsourcing experiments. Additionally, this could support designers or operators of narrative-driven recommender systems, for example, in online discussion forums to automatically compute suggestions instantly without any language based restrictions, such as defining a specific language (e.g., bot language) when writing a narrative.

Recommender Bot. Another interesting part for future work would be to conduct experiments on online discussion boards by implementing a recommender bot that users can query for recommendations while providing a narrative. Further, an implemented feedback loop, for example, by taking into account up- and down-votes for individual suggestions from users, could improve future bot recommendations. Such a bot could help to evaluate the importance of additional metrics, such as diversity, serendipity, or novelty in the context of narrative-driven recommendations.

Overall, I believe that this thesis serves as a stepping stone towards further research in the direction of recommender systems utilizing social as well as narrative features.

List of Figures

2.1 Matrix Factorization 23
2.2 Arithmetic Operations with Word Embeddings 27

3.1 Distributions of Purchases, Sellers, and Buyers 41
3.2 Mean Fraction of Sellers 46
3.3 Training and Test Set Sizes 47
3.4 Individual Feature Performances. 49
3.5 Feature Performances over Time 50
3.6 Information Gain of Features 88
3.7 Feature Correlations 89
3.8 Distributions of Movies in Requests and Suggestions . . . 105
3.9 Frequency of Genres 107
3.10 Word Clouds of Positive and Negative Keywords 108
3.11 Overlap of Movies and Keywords in Requests and Suggestions 111
3.12 Feature Importance 124
3.13 Experimental Setup 137
3.14 Results 147
3.15 Score Weights 149
3.16 Empirical Recommendations 152

List of Tables

1.1 Overview of the Main Publications 13

2.1 Item-Item Similarity 22

3.1 Marketplace Dataset Statistics 40
3.2 Online Social Network Dataset Statistics 42
3.3 Overview of Related Work 56
3.4 Extracted Entities 66
3.5 Statistics of the Online Social Network G_O 68
3.6 Statistics of the Location-Based Social Network G_L 70
3.7 Statistics of the Trading Network G_T 71
3.8 Statistics of the Combined Network G_C 72
3.9 Overview of all Features 82
3.10 Individual Feature Performances 87
3.11 Feature Set Performances 90
3.12 Reddit Dataset Characteristics 104
3.13 IMDb Dataset Characteristics 116
3.14 Graph Statistics 117
3.15 Hyperparameters 119
3.16 Best Parameter Configurations 120
3.17 Reddit Evaluation Dataset Characteristics 134
3.18 IMDb Dataset Characteristics 139
3.19 Evaluation Protocol 146

List of Boxes

1.1 Request and Suggestions Example 4

3.1 Request and Suggestions Example 99
3.2 Request Examples 115

Bibliography

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3):211–230.

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1):103–145.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Adomavicius, G. and Tuzhilin, A. (2011). *Context-Aware Recommender Systems*, pages 217–253. Springer US, Boston, MA.

Adomavicius, G., Tuzhilin, A., and Zheng, R. (2011). Request: A query language for customizing recommendations. *Information Systems Research*, 22(1):99–117.

Al Hasan, M. and Zaki, M. J. (2011). A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. Springer.

Atzmueller, M. and Roth-Berghofer, T. (2010). The Mining and Analysis Continuum of Explaining Uncovered. In *Proc. 30th SGAI International Conference on Artificial Intelligence (AI-2010)*.

Backstrom, L. and Leskovec, J. (2011). Supervised Random Walks: Predicting and Recommending Links in Social Networks. In *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM)*, pages 635–644.

Balby Marinho, L., Trattner, C., and Parra, D. (2015). Are real-world place recommender algorithms useful in virtual world environments? In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 245–248.

Barabasi, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512.

Barabasi, A.-L., Jeong, H., Neda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311:590–614.

Barbieri, N., Bonchi, F., and Manco, G. (2014). Who to Follow and Why: Link Prediction with Explanations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1266–1275. ACM.

Barkan, O. and Koenigstein, N. (2016). Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Basu, C., Hirsh, H., Cohen, W., et al. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M., and Blackburn, J. (2020). The pushshift reddit dataset. *arXiv preprint arXiv:2001.08435*.

Baumgartner, J. M. (2015). Reddit comment dataset. Website. https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicly_available_reddit_comment.

Beyer, K. S., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When Is "Nearest Neighbor" Meaningful? In *Proceedings of the 7th International Conference on Database Theory*, pages 217–235. Springer-Verlag.

Biran, O. and Cotton, C. (2017). Explanation and Justification in Machine Learning: A Survey. In *IJCAI-17 Workshop on Explainable AI*.

Bischoff, K. (2012). We love rock 'n' roll: analyzing and predicting friendship links in Last.fm. In *Proceedings of the 3rd Annual ACM Web Science Conference*, pages 47–56. ACM.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bogers, T. (2015). Searching for movies: An exploratory analysis of movie-related information needs. *iConference 2015 Proceedings*.

Bogers, T. and Koolen, M. (2017). Defining and supporting narrative-driven recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 238–242. ACM.

Brin, S. and Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30(1-7):107–117.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Cai, L., Xu, J., Liu, J., and Pei, T. (2018). Integrating spatial and temporal contexts into a factorization model for poi recommendation. *International Journal of Geographical Information Science*, 32(3):524–546.

Campos, P., Diaez, F., and Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119.

Chakrabarti, S. (2007). Dynamic personalized pagerank in entity-relation graphs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 571–580, New York, NY, USA. ACM.

Chen, J., Wu, Y., Fan, L., Lin, X., Zheng, H., Yu, S., and Xuan, Q. (2017). Improved spectral clustering collaborative filtering with node2vec technology. In *2017 International Workshop on Complex Systems and Networks (IWCSN)*, pages 330–334. IEEE.

Chen, J., Wu, Y., Fan, L., Lin, X., Zheng, H., Yu, S., and Xuan, Q. (2019). N2vscdnnr: A local recommender system based on node2vec and rich information network. *IEEE Transactions on Computational Social Systems*, 6(3):456–466.

Cheng, J., Romero, D. M., Meeder, B., and Kleinberg, J. (2011). Predicting Reciprocity in Social Networks. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 49–56.

Christakopoulou, K., Radlinski, F., and Hofmann, K. (2016). Towards conversational recommender systems. In *KDD*, pages 815–824.

Christakou, C., Vrettos, S., and Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, 16(05):771–792.

Coleman, J. S. (1988). Social Capital in the Creation of Human Capital. *American Journal of Sociology*, pages 95–120.

Cranshaw, J., Toch, E., Hong, J., Kittur, A., and Sadeh, N. (2010). Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM.

da Silva Soares, P. R. and Prudêncio, R. B. C. (2012). Time series based link prediction. In *The 2012 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE.

Delporte, J., Karatzoglou, A., Matuszczyk, T., and Canu, S. (2013). Socially enabled preference learning from implicit feedback data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–160. Springer.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pages 107–144.

DiMaggio, P. and Louch, H. (1998). Socially embedded consumer transactions: For what kinds of purchases do people most often use networks? *American sociological review*, pages 619–637.

Eberhard, L. and Trattner, C. (2016). Recommending sellers to buyers in virtual marketplaces leveraging social information. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, page 559–564, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Eberhard, L., Trattner, C., and Atzmueller, M. (2019a). Predicting trading interactions in an online marketplace through location-based and online social networks. *Information Retrieval Journal*, 22(1-2):55–92.

Eberhard, L., Walk, S., and Helic, D. (2020). Tell me what you want: Embedding narratives for movie recommendations. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20, page 301–306, New York, NY, USA. Association for Computing Machinery.

Eberhard, L., Walk, S., Posch, L., and Helic, D. (2019b). Evaluating narrative-driven movie recommendations on reddit. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 1–11, New York, NY, USA. ACM.

Eirinaki, M., Gao, J., Varlamis, I., and Tserpes, K. (2018). Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions.

Eisner, B., Rocktäschel, T., Augenstein, I., Bošnjak, M., and Riedel, S. (2016). emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Elsafty, A., Riedl, M., and Biemann, C. (2018). Document-based recommender system for job postings using dense representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 216–224, New Orleans - Louisiana. Association for Computational Linguistics.

Feng, W. and Wang, J. (2012). Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1276–1284.

Fire, M., Tenenboim, L., Lesser, O., Puzis, R., Rokach, L., and Elovici, Y. (2011). Link Prediction in Social Networks using Computationally Efficient Topological Features. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust (PASSAT) / 2011 IEEE Third Int'l Conference on Social Computing (SocialCom)*, pages 73–80. IEEE.

Fire, M., Tenenboim-Chekina, L., Puzis, R., Lesser, O., Rokach, L., and Elovici, Y. (2013). Computationally Efficient Link Prediction in a Variety of Social Networks. *ACM TIST*, 5(1):10:1–10:25.

Funk, S. (2006). Netflix update: Try this at home.

Getoor, L. and Diehl, C. P. (2005). Link Mining: A Survey. *Acm Sigkdd Explorations Newsletter*, 7(2):3–12.

Ghosh, S., Mundhe, M., Hernandez, K., and Sen, S. (1999). Voting for movies: The anatomy of a recommender system. In *Proceedings of the Third Annual Conference on Autonomous Agents*, AGENTS '99, pages 434–435, New York, NY, USA. ACM.

Glenski, M. and Weninger, T. (2017). Predicting user-interactions on reddit. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 609–612.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., and Sharp, D. (2015). E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1809–1818. ACM.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Guo, S., Wang, M., and Leskovec, J. (2011). The role of social networks in online shopping: information passing, price of trust, and consumer choice. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 157–166.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Hariri, N., Mobasher, B., and Burke, R. (2013). Query-driven context aware recommendation. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 9–16, New York, NY, USA. ACM.

Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *Proceedings of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Huang, Z., Li, X., and Chen, H. (2005). Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '05, pages 141–142, New York, NY, USA. ACM.

Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 135–142, New York, NY, USA. ACM.

Jones, J. J., Settle, J. E., Bond, R. M., Fariss, C. J., Marlow, C., and Fowler, J. H. (2013). Inferring Tie Strength from Online Directed Behavior. *PLoS ONE*, 8(1).

Kallumadi, S. and Hsu, W. H. (2018). Interactive recommendations by combining user-item preferences with linked open data. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, pages 121–125.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43.

Kibanov, M., Atzmueller, M., Illig, J., Scholz, C., Barrat, A., Cattuto, C., and Stumme, G. (2015). Is Web Content a Good Proxy for Real-Life Interaction? A Case Study Considering Online and Offline Interactions of Computer Scientists. In *Proc. ASONAM*, Boston, MA, USA. IEEE Press.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM.

Kranton, R. E. and Minehart, D. F. (2001). A Theory of Buyer-Seller Networks. *American economic review*, 91(3):485–508.

La, C.-A. and Michiardi, P. (2008). Characterizing User Mobility in Second Life. In *Proceedings of the first workshop on Online social networks*, WOSN '08. ACM.

Lacic, E., Kowald, D., Eberhard, L., Trattner, C., Parra, D., and Marinho, L. B. (2015). Utilizing online social network and location-based data to recommend products and categories in online marketplaces. In *Mining, Modeling, and Recommending 'Things' in Social Media*, pages 96–115, Cham. Springer International Publishing.

Lamprecht, D., Geigl, F., Karas, T., Walk, S., Helic, D., and Strohmaier, M. (2015). Improving recommender system navigability through diversification: A case study of imdb. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, i-KNOW '15, pages 21:1–21:8, New York, NY, USA. ACM.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Lee, G. M., Qiu, L., and Whinston, A. B. (2016). A friend like me: modeling network formation in a location-based social network. *Journal of Management Information Systems*, 33(4):1008–1033.

Lehdonvirta, V. (2009). Virtual item sales as a revenue model: identifying attributes that drive purchase decisions. *Electronic commerce research*, 9(1):97–113.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 641–650. ACM.

Levene, H. (1960). *In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling, I. Olkin et al. eds*, pages 278–292. Stanford University Press.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Li, X. and Huan, J. (2017). Constructivism Learning: A Learning Paradigm for Transparent Predictive Analytics. In *Proc. SIGKDD*, pages 285–294. ACM.

Liben-Nowell, D. and Kleinberg, J. (2003). The link-prediction problem for social networks. In *Conference on Information and Knowledge Management (CIKM'03)*, pages 556–559.

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

Lichtenwalter, R. and Chawla, N. V. (2011). Lpmade: Link prediction made easy. *Journal of Machine Learning Research*, 12:2489–2492.

Lichtenwalter, R. and Chawla, N. V. (2012). Vertex collocation profiles: subgraph counting for link analysis and prediction. In *WWW*, pages 1019–1028.

Lichtenwalter, R., Lussier, J. T., and Chawla, N. V. (2010). New Perspectives and Methods in Link Prediction. In *KDD*, pages 243–252.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

Liu, D.-R., Chou, Y.-C., Chung, C.-C., and Liao, H.-Y. (2018). Recommender System based on Social Influence and the Virtual House Bandwagon Effect in Virtual Worlds. *Kybernetes*, 47(3):587–604.

Lü, L. and Zhou, T. (2010). Link prediction in weighted networks: The role of weak ties. *EPL (Europhysics Letters)*, 89(1):18001.

Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170.

Lu, Y., Tsaparas, P., Ntoulas, A., and Polanyi, L. (2010a). Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World wide web*, pages 691–700.

Lu, Z., Savas, B., Tang, W., and Dhillon, I. S. (2010b). Supervised Link Prediction Using Multiple Sources. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 923–928. IEEE.

Ma, H., Zhou, D., Liu, C., Lyu, M. R., and King, I. (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Macedo, A. Q., Marinho, L. B., and Santos, R. L. (2015). Context-aware event recommendation in event-based social networks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 123–130.

Mahmood, T. and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 73–82, New York, NY, USA. ACM.

Mak, H., Koprinska, I., and Poon, J. (2003). Intimate: A web-based movie recommender using text categorization. In *Web Intelligence,* *2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 602–605. IEEE.

Manotumruksa, J., Macdonald, C., and Ounis, I. (2016). Modelling user preferences using word embeddings for context-aware venue recommendation. *arXiv preprint arXiv:1606.07828*.

Martin, T. (2017). community2vec: Vector representations of online communities encode semantic relationships. In *Proceedings of the Second Workshop on NLP and Computational Social Science*, pages 27–31, Vancouver, Canada. Association for Computational Linguistics.

Martínez, V., Berzal, F., and Cubero, J.-C. (2016). A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):1–33.

McGinty, L. and Reilly, J. (2011). On the evolution of critiquing recommenders. In *Recommender Systems Handbook*, pages 419–453. Springer.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Riew of Sociology*, 27(1):415–444.

Menon, A. K. and Elkan, C. (2011). Link Prediction via Matrix Factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 437–452. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746–751.

Mnih, A. and Hinton, G. E. (2009). A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088.

Murata, T. and Moriyasu, S. (2007). Link Prediction of Social Networks Based on Weighted Proximity Measures. In *Web Intelligence*, pages 85–88.

Musto, C., Basile, P., and Semeraro, G. (2019). Embedding knowledge graphs for semantics-aware recommendations based on dbpedia. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 27–31.

Musto, C., Semeraro, G., De Gemmis, M., and Lops, P. (2015). Word embedding techniques for content-based recommender systems: An empirical evaluation. In *RecSys Posters*.

Newman, M. (2018). *Networks*. Oxford university press.

Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*, 67(2):026126.

Newman, M. E. J. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102.

Nunes, I. and Jannach, D. (2017). A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, pages 1–52.

Oku, K., Nakajima, S., Miyazaki, J., and Uemura, S. (2006). Context-aware SVM for context-dependent information recommendation. In *Proceedings of the 7th international Conference on Mobile Data Management*, page 109. IEEE Computer Society.

Ozsoy, M. G. (2016). From word embeddings to item recommendation. *arXiv preprint arXiv:1601.01356*.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. (2009). Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, pages 265–268. ACM.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8.

Pearson, K. (1905). The problem of the random walk. *Nature*, 72(1865):294.

Perny, P. and Zucker, J.-D. (2001). Preference-based search and machine learning for collaborative filtering: the "film-conseil" movie recommender system. *Information, Interaction, Intelligence*, 1(1):9–48.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA. ACM.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *Proc. ACM SIGKDD*, pages 1135–1144. ACM.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

Ripley, R. M., Snijders, T. A., Boda, Z., Vörös, A., and Preciado, P. (2011). Manual for RSIENA. *University of Oxford, Department of Statistics, Nuffield College*, 1.

Rowe, M., Stankovic, M., and Alani, H. (2012). Who will follow whom? Exploiting Semantics for Link Prediction in Attention-Information Networks. In *Proceedings of the 11th international conference on The Semantic Web - Volume Part I*, ISWC'12, pages 476–491. Springer-Verlag.

Salakhutdinov, R. and Mnih, A. (2007). Probabilistic matrix factorization. In *Nips*, volume 1, pages 2–1.

Salakhutdinov, R. and Mnih, A. (2008). Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Salton, G. and McGill, M. J. (1986). Introduction to modern information retrieval.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA. ACM.

Scellato, S., Noulas, A., and Mascolo, C. (2011). Exploiting Place Features in Link Prediction on Location-based Social Networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM.

Schafer, J., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer Heidelberg.

Schafer, J. B., Konstan, J., and Riedl, J. (1999). Recommender Systems in E-Commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166. ACM.

Scholz, C., Atzmueller, M., Barrat, A., Cattuto, C., and Stumme, G. (2013a). New Insights and Methods For Predicting Face-To-Face Contacts. In *Proc. 7th Intl. AAAI Conference on Weblogs and Social Media*.

Scholz, C., Atzmueller, M., Kibanov, M., and Stumme, G. (2013b). How Do People Link? Analysis of Contact Structures in Human Face-to-Face Proximity Networks. In *Proc. ASONAM 2013*, New York, NY, USA. ACM Press.

Scholz, C., Atzmueller, M., and Stumme, G. (2012). On the Predictability of Human Contacts: Influence Factors and the Strength of Stronger Ties. In *SocialCom 2012*, Los Alamitos, CA, USA. IEEE Computer Society.

Scholz, C., Illig, J., Atzmueller, M., and Stumme, G. (2014). On the Predictability of Talk Attendance at Academic Conferences. In *Proc. 25th ACM Conference on Hypertext and Social Media*, New York, NY, USA. ACM Press.

Seitlinger, P., Kowald, D., Kopeinik, S., Hasani-Mavriqi, I., Ley, T., and Lex, E. (2015). Attention please! a hybrid resource recommender mimicking attention-interpretation dynamics. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 339–345, New York, NY, USA. ACM.

Spitzer, F. (2013). *Principles of random walk*, volume 34. Springer Science & Business Media.

Steurer, M. and Trattner, C. (2013a). Acquaintance or partner? predicting partnership in online and location-based social networks. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 372–379. IEEE.

Steurer, M. and Trattner, C. (2013b). Predicting Interactions in Online Social Networks: An Experiment in Second Life. In *Proceedings of the 4th International Workshop on Modeling Social Media*, pages 5:1–5:8. ACM.

Steurer, M. and Trattner, C. (2013c). Who will Interact with Whom? A Case-Study in Second Life Using Online Social Network and Location-Based Social Network Features to Predict Interactions between Users. In *Ubiquitous Social Media Analysis*, volume 8329 of *Lecture Notes in Computer Science*, pages 108–127. Springer Berlin Heidelberg.

Steurer, M., Trattner, C., and Helic, D. (2013). Predicting Social Interactions from Different Sources of Location-based Knowledge. In *The Third International Conference on Social Eco-Informatics*, SOTICS 2013, pages 8–13, Lisbon, Portugal.

Stiebellehner, S., Wang, J., and Yuan, S. (2018). Learning Continuous User Representations through Hybrid Filtering with doc2vec. *CoRR*, abs/1801.0.

Suh, B., Convertino, G., Chi, E. H., and Pirolli, P. (2009). The singularity is not near: slowing growth of wikipedia. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, pages 1–10.

Szell, M., Sinatra, R., Petri, G., Thurner, S., and Latora, V. (2012). Understanding mobility in a social petri dish. *Scientific reports*, 2.

Terveen, L. and Hill, W. (2001). Beyond recommender systems: Helping people help each other. *HCI in the New Millennium*, 1(2001):487–509.

Thelwall, M. (2009). Homophily in MySpace. *Journal of the American Society for Information Science and Technology*, 60(2):219–231.

Thiele, L., Atzmueller, M., Stumme, G., and Kauffeld, S. (2018). Frequently and/or Durable? The Predictive Impact of Initial Face-to-Face Contacts on the Formation and Evolution of Developmental Peer Network Relationships. *Psychology.*

Thorelli, H. B. (1986). Networks: Between Markets and Hierarchies. *Strategic management journal*, 7(1):37–51.

Tintarev, N. and Masthoff, J. (2007). A survey of explanations in recommender systems. In *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, pages 801–810. IEEE.

Trattner, C., Parra, D., Eberhard, L., and Wen, X. (2014). Who will trade with whom? predicting buyer-seller interactions in online trading platforms through social networks. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, page 387–388, New York, NY, USA. Association for Computing Machinery.

Trattner, C. and Steurer, M. (2015). Detecting partnership in location-based and online social networks. *Social Network Analysis and Mining*, 5(1).

Tylenda, T., Angelova, R., and Bedathur, S. (2009). Towards time-aware link prediction in evolving social networks. In *Proceedings of the 3rd workshop on social network mining and analysis*, pages 1–10.

Varvello, M., Picconi, F., Diot, C., and Biersack, E. (2008). Is There Life in Second Life? In *Conext'08*, Madrid, Spain.

Varvello, M. and Voelker, G. M. (2010). Second Life: A Social Network of Humans and Bots. In *Proceedings of the 20th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '10, pages 9–14. ACM.

Wang, P., Xu, B., Wu, Y., and Zhou, X. (2015). Link Prediction in Social Networks: the State-of-the-Art. *Science China Information Sciences*, 58(1):1–38.

Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM.

Yang, J., Wei, X., Ackerman, M. S., and Adamic, L. A. (2010). Activity lifespan: An analysis of user survival patterns in online knowledge sharing communities. In *Fourth International AAAI Conference on Weblogs and Social Media*.

Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008a). A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610. ACM.

Yilmaz, E., Kanoulas, E., and Aslam, J. A. (2008b). A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 603–610, New York, NY, USA. ACM.

Yue, Y., Finley, T., Radlinski, F., and Joachims, T. (2007). A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 271–278, New York, NY, USA. ACM.

Zhang, J., Kong, X., and Philip, S. Y. (2013). Predicting social links for new users across aligned heterogeneous social networks. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1289–1294. IEEE.

Zhang, J. and Philip, S. Y. (2014). Link Prediction across Heterogeneous Social Networks: A Survey. *SOCIAL NETWORKS*.

Zhang, Y. and Pennacchiotti, M. (2013). Predicting Purchase Behaviors from Social Media. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1521–1532.

Zhuang, H., Chin, A., Wu, S., Wang, W., Wang, X., and Tang, J. (2012a). Inferring Geographic Coincidence in Ephemeral Social Networks. In *ECML/PKDD*. Springer.

Zhuang, H., Tang, J., Tang, W., Lou, T., Chin, A., and Wang, X. (2012b). Actively Learning to Infer Social Ties. *Data Min. Knowl. Discov.*, 25(2):270–297.