11-2018

Analyzing and modeling users in multiple online social platforms

Roy LEE KA WEI
Singapore Management University, roylee.2013@phdis.smu.edu.sg

ANALYZING AND MODELING USERS IN MULTIPLE ONLINE SOCIAL PLATFORMS

ROY KA-WEI LEE

SINGAPORE MANAGEMENT UNIVERSITY

2018

Analyzing and Modeling Users in Multiple Online Social Platforms

Roy Ka-Wei Lee

Submitted to School of Information Systems

in partial fulfillment of the requirements for the

Degree of Doctor of Philosophy in Information Systems

Dissertation Committee:

Lim Ee-Peng (Supervisor/Chair)
Professor of Information Systems
Singapore Management University

Zheng Baihua
Associate Professor of Information Systems
Singapore Management University

David Lo
Associate Professor of Information Systems
Singapore Management University

Teow Loo Nin
Distinguish Member of Technical Staff
DSO National Laboratories

Singapore Management University
2018

I hereby declare that this PhD dissertation is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree in any university previously.

Roy Ka-Wei Lee

30 November 2018

Analyzing and Modeling Users in Multiple Online Social Platforms

by

Roy Ka-Wei Lee

Abstract

This dissertation addresses the empirical analysis on user-generated data from multiple online social platforms (OSPs) and modeling of latent user factors in multiple OSPs setting.

In the first part of this dissertation, we conducted cross-platform empirical studies to better understand user's social and work activities in multiple OSPs. In particular, we proposed new methodologies to analyze users' friendship maintenance and collaborative activities in multiple OSPs. We also apply the proposed methodologies on real-world OSP datasets, and the findings from our empirical studies have provided us with a better understanding on users' social and work activities which are previously not uncovered in single OSP studies.

In the second part of this dissertation, we developed user modeling techniques to learn latent user factors in multiple OSPs setting. In particular, we proposed generative models to learn the user topical interests, topic-specific platform preferences and influences in multiple OSPs setting. The proposed models are also applied to real-world OSPs datasets to profile user topical interests and identify influential users in multiple OSPs. The designed generative models are also generalizable and can be applied to different cross-OSP datasets.

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation is impossible without the help from my advisor, Prof. Lim Ee-Peng. Prof. Lim has been an inspiration to me; he has taught me many skills that have benefited me throughout my Ph.D. journey and in my future career. I want to thank him for his guidance, patience, and encouragement that made my Ph.D. experience fun and productive.

I am thankful to my dissertation committee members: Prof. Zhang Baihua, Prof. David Lo, and Dr. Teow Loo Nin. They have provided helpful suggestions and insightful comments to improve this dissertation. In particular, I would also like to thank Prof. David for his guidance and research input that inspired the work in Chapter 4 in this dissertation.

During my Ph.D. study, I was fortunate to have the consistent support and encouragement from my friends and colleagues at Singapore Management University and Living Analytics Research Centre: Dr. Chong Wen Haw, Dr. Chiang Meng-Fen, Dr. Richard Oentaryo, Dr. Do Ha Loc, Dr. Xie Wei, Larry Lin, James Hoang, and Hee Mingshan. A big thank you to all of you, for making my Ph.D. journey a colorful and fun one. I would also like to especially thank Dr. Tuan-Anh Hoang, my senior and collaborator, for sharing his experience and knowledge. The topic modeling techniques that I have learned from him laid the foundation for the work in the second part of this dissertation.

I would also like to thank Seow Pei Huan, Phoebe Yeo, Jamie Chia, Philips Prasetyo, Fong Soon Keat, and Desmond Yap for their help and support in

administrative and technical matters.

Finally, I would like to thank my mother and grandparents for their love and strong support during my graduate study. This dissertation is dedicated to them.

Dedicated to my family and friends

Publications

Publications based on the dissertation:

1. Roy Ka-Wei Lee, Tuan-Anh Hoang and Ee-Peng Lim, *Discovering Hidden Topical Hubs and Authorities in Online Social Networks*, SIAM Internation Conference on Data Mining (SDM), 2018.
2. Roy Ka-Wei Lee and David Lo, *Wisdom in Sum of Parts: Multi-Platform Activity Prediction in Social Collaborative Sites*, ACM Conference on Web Science (WebSci), 2018.
3. Roy Ka-Wei Lee, Tuan-Anh Hoang and Ee-Peng Lim, *On Analyzing User Topic-Specific Platform Preferences Across Multiple Social Media Sites*, World Wide Web Conference (WWW), 2017.
4. Roy Ka-Wei Lee and David Lo, *GitHub and Stack Overflow: Analyzing Developer Interests Across Multiple Social Collaborative Platforms*, International Conference on Social Informatics (SocInfo), 2017.
5. Roy Ka-Wei Lee and Ee-Peng Lim, *Friendship Maintenance and Prediction in Multiple Social Networks*, ACM Conference on Hypertext and Social Media (HT), 2016.

Manuscript based on the dissertation and under review:

1. Roy Ka-Wei Lee, Tuan-Anh Hoang and Ee-Peng Lim, *Discovering Hidden Topical Hubs and Authorities Across Multiple Online Social Networks*, submitted to IEEE Transaction on Knowledge and Data Engineering (TKDE).

Other publications not included in dissertation.

1. Roy Ka-Wei Lee and Ee-Peng Lim, *Measuring User Influence, Susceptibility and Cynicalness in Sentiment Diffusion*, European Conference on Information Retrieval (ECIR), 2015.

Chapter 1

Introduction

1.1 Motivation

With the proliferation of online social platforms (OSPs), users today find themselves engaging and connecting with each other on OSPs [24]. For example, users may "like" the posts of their friends on Facebook, *retweet* users whom they have followed on Twitter or share photos on Instagram. Besides engaging each other in social activities, users also leverage on OSPs for collaborative works. For example, software engineers have used social collaborative platforms such as GitHub, a platform that allows sharing of software codes with other users, and Stack Overflow, a community-based website for asking and answering questions relating to software engineering, for software development [15, 70].

The users' participation in multiple OSPs generates voluminous and rich data about the users. Some of these user-generated data include:

- **Profile Attributes**: These are attributes that describe a user's profile in an OSP. Examples of such attributes include username, short bio description, etc.

- **Activities**: These are social and work activities performed by users in OSPs. An example of a social activity is the like a Facebook user gives to

some post. In Stack Overflow, a user-answer-question is a work activity example.

- **Relationships**: These are the directed or undirected connections between users in OSPs. Examples include users *follow* other users in Twitter and *friends* between users in Facebook. These connections serve either social or information purposes. Social connections are meant for users to establish friendships. Information connections, in contrast, are meant for receiving content of interests to the users.

- **Content**: These are media content generated by users for self-journaling or sharing. The media content may exist in the text form (e.g., tweets in Twitter) or multimedia form (e.g., photos in Instagram).

Analyzing and modeling these user-generated data across multiple OSPs are essential tasks in many real-world applications. Firstly, a multi-platform approach to analyze user-generated data allows us to profile users more effectively. Consider a user who publishes political-related posts and follows politicians in Twitter. Suppose the user also publishes music-related posts in Tumblr. Based on the user's tweet data only, one could infer his interests in politics but not music. By profiling user interest using both his Twitter and Tumblr data, such a drawback can be avoided. Moreover, one can learn the platform preferences of users as they decide to share content and interact with others. Secondly, the multi-platform approach also enables us to build better recommender systems. For example, when an active Facebook user joins Instagram, we can recommend her to connect with her friends on Facebook who also have accounts on Instagram or recommend her topic-specific influential users to follow in Instagram considering her topical interests across platforms. We can also make other similar recommendations when users are active on multiple social collaborative platforms. For instance, if a user commits to a Java-related repository in GitHub, we can recommend her Java-related ques-

tions to answer in Stack Overflow. The above scenarios assume that the users act similarly in different OSPs. When the assumption no longer holds, it is crucial to learn the behavioral characteristics of individual users in different OSPs for better recommendations.

1.1.1 Challenges

Despite the many real-world applications, there are also challenges in the analysis and modeling of user-generated data across multiple OSPs.

Lack of cross-platform datasets. While there are abundant user-generated data from multiple OSPs, there are little information on user- user linkage, i.e., not many users declare the different OSP accounts they own. Consequently, it is difficult to collect cross-platform datasets where multiple OSP accounts belonging to the same user are linked. To tackle this challenge, many user identity linkage methods to find the OSP accounts belonging to the same user have been proposed [136, 137, 75, 60, 92, 121, 88, 27]. As this itself is a vibrant research topic, this dissertation does not seek to address it but instead uses the methods to match user accounts before we apply our analysis and modeling techniques to the user-generated data from multiple OSPs.

Lack of analysis and modeling techniques. Many of the previous research work on user activities, relationships, and content in OSPs, are restricted to the single OSP setting. Their analysis and modeling techniques are inadequate when applied to multiple OSPs. Firstly, there are new latent user factors unique to multiple OSPs setting which existing single OSP methods are unable to model and analyze. For example, users who have accounts on multiple OSPs may have platform preferences when posting content or forming relationships, which could not be modeled by single OSP methods. The platform preferences may also be specific to the certain activity, relationship and content types. For example, a user may prefer to maintain friendship in Facebook while keeping work relationships (e.g., colleagues) in LinkedIn. Secondly,

there is a need to match the different types of users' activities, relationships and content across multiple OSPs to perform comparative studies and cross-platform analysis. For example, Twitter has *follow* relationship while Facebook has *friendship* between users. Thus, new methods will need to be proposed to match the different relationship types in Twitter and Facebook before analyzing the cross-platform social relationships of users who have accounts on Twitter and Facebook. Furthermore, the learning of such multiple OSPs latent user factors is also non-trivial as it is difficult to model their interactions leading to the observed user-generated data.

Lack of ground truth. The lack of ground truth labels used for evaluating latent user factors in OSP is a known challenge even for studies in single OSP setting. However, the analysis and modeling in multiple OSP setting complicate it further because of (i) the introduction of new latent user factors, and (ii) the interaction between new factors and other latent user factors studied in single OSP setting. For example, on identifying influential users across multiple OSPs, it is difficult to evaluate the influence of users because the ground truth is not available. New evaluation approaches and synthetic datasets need to be proposed to overcome this limitation, and such tasks are also non-trivial.

1.2 Research Objectives

In this dissertation, we aim to address the user-generated data analysis and modeling gaps and challenges in multiple OSPs research by adopting the research framework shown in Figure 1.1. The research framework consists of three inter-linked components, namely: (i) user-generated data, (ii) empirical analysis, and (iii) user modeling.

User-generated data. We aim to collect user-generated data from users who have accounts on multiple OSPs. To achieve this goal, we will first gather

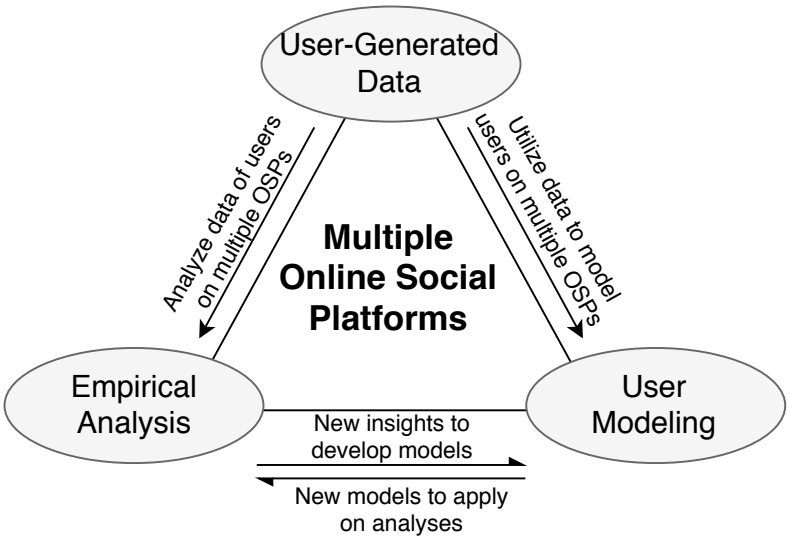


Figure 1.1: Research Framework

users and their profile attributes such as a short biography description of a specific platform. From their short biography descriptions, we will identify users who have declared their user accounts in other OSPs. Subsequently, we will collect the user-generated data such as relationship and content of the identified user accounts in multiple OSPs using the respective OSP's APIs. As there may not be many users who declared their accounts in multiple OSPs, existing state-of-the-art user profile linkage techniques can also be applied to find the OSP accounts belonging to the same user. The collected user-generated data will be used in cross-platform empirical studies and user modeling tasks.

Empirical analysis. We aim to conduct cross-platform empirical studies on the user-generated data collected from multiple OSPs. In particular, we will focus on analyzing and comparing how users manage their activities, relationships, content across multiple OSPs. For example, we can study how users distribute and maintain their social relationships across different OSPS. The complementary and substitution relations between OSPs may also be explored. For example, we can analyze how a user performs substituting or complementing work activities in different OSPs to achieve a specific task. As the existing methods proposed on single OSP setting are not able to perform such cross-platform analysis, novel methods will also be introduced to ana-

lyze user-generated data in the multiple OSP setting. The insights gathered from the cross-platform empirical studies may be used in designing user modeling tasks. For example, after analyzing how a user performs work activities in different OSPs, we can develop methods to model the latent users' work preferences in multiple OSP setting.

User modeling. We aim to model latent user factors in multiple OSPs setting. To achieve this goal, we will propose new methods to handle heterogeneous user-generated data from multiple OSPs. Also, novel modeling techniques will also be proposed to learn new latent user factors unique to multiple OSPs context. For example, a user may prefer to post about music-related topics in an OSP while sharing politics-related posts in another OSP. This topic-specific platform preference is unique to multiple OSP setting, and novel modeling techniques will need to be proposed to learn such latent user factors. The proposed user modeling techniques and methodologies can also be utilized in the cross-platform empirical analysis. For instance, we apply the model which learns the users' topic-specific platform preference to study the similarity between user topical interests in multiple OSPs.

1.3 Contributions

In this dissertation, we aim to contribute to the state-of-the-art by conducting two empirical studies on user-generated data in multiple OSPs and performing two user modeling tasks to learn latent user factors in multiple OSP setting.

1.3.1 Empirical Studies

We aim to conduct empirical studies to better understand user's social and work behaviors in multiple OSPs. In particular, we study users' (i) friendship maintenance and (ii) collaborative activities in multiple OSPs.

Empirical Study 1. On analyzing user friendships in multiple OSPs, we

propose novel measures that quantify the similarity and evenness of a user's friendship in multiple OSPs. These measures will be used to empirically analyze the friendships of users who have accounts on Twitter and Instagram. Interestingly, we find that most users prefer to maintain different friendships in Twitter and Instagram while keeping only a small clique of common friends across the two OSPs. Also, most users prefer to have roughly the same number of friends in the two OSPs. The findings from our empirical study provide us with a better understanding of users' social activities and relationships which are previously not uncovered in single OSP studies. The insights from our user's friendship maintenance empirical study can also be used to derive novel user features which can improve friendship link prediction in multiple OSP setting.

Empirical Study 2. On analyzing collaborative activities in multiple OSPs, we propose novel measures to quantify the similarity in users' topical interests inferred from their collaborative activities within and across multiple OSPs. We collect large datasets from GitHub and Stack Overflow, which are two popular OSPs used by the software engineering communities for collaborative works and apply our proposed measures to study the users' collaborative activities in the two OSPs. Interestingly, we find that users with accounts on GitHub and Stack Overflow do display some similar topical interests in their collaborative activities across the OSPs. Furthermore, users share similar topical interests with other users who perform collaborative activities together in the two OSPs. We also demonstrate that we able to predict a user's collaborative activities in one OSP (e.g., GitHub) using the same user's topical interests inferred from his or her collaborative activities in another OSP (e.g., Stack Overflow). Our empirical study is the first work that study users' topical interests across GitHub and Stack Overflow and the findings from our study help to better on understanding user's work activities in multiple OSPs.

1.3.2 User Modeling Tasks

We aim to model the latent user factors in multiple OSP setting considering the platform-specificity of the latent user factors. In particular, we model the (i) user topical interests and (ii) user influences in multiple OSP setting.

Modeling Task 1. On modeling user topical interests in multiple OSPs, we propose MultiPlatform-LDA (MultiLDA), which is a generative model that learns users' latent topical interests in multiple OSPs and their topic-specific platform preferences. Through experiments on real-world datasets, we show that MultiLDA can model user topical interests in multiple OSPs and we demonstrate the predictive power of our model by predicting the platform which a user will publish for a given generated post by him or her. Empirically, we apply MultiLDA to learn user topics on Twitter, Instagram, and Tumblr. The proposed MultiLDA model can improve user profiling in multiple OSPs.

Modeling Task 2. On modeling user influence in multiple OSPs, we propose two novel generative models, Hub and Authority Topic model (HAT) and Multiple Platform Hub and Authority Topic model (MPHAT), to identity topic-specific influential users in single and multiple OSP settings. We apply HAT and MPHAT on real-world datasets and demonstrate that HAT and MPHAT performed well in (a) topic modeling, (b) platform prediction, and (c) user link recommendation, for both single and multiple OSP settings. Empirically, we also show that HAT and MPHAT can identify topic-specific hubs and authorities within and across Instagram and Twitter. The proposed HAT and MPHAT models can improve recommendation systems in OSPs.

1.4 Organization of the Dissertation

The rest of this dissertation is structured as follows. Chapter 2 surveys related work. We present our empirical studies on user friendships and collaborative activities in Chapter 3 and 4 respectively. In Chapter 5, we present the Mul-

tiLDA model for learning user topical interests and platform preferences in multiple OSPs. The HAT and MPHAT models for learning topic-specific hub and authority users in single and multiple OSPs are described in Chapter 6. Finally, we conclude this dissertation and discuss some directions for future work in Chapter 7.

Chapter 2

Related Work

In this chapter, we survey five threads of previous literature that are closely related to this dissertation research and highlight the differences between our works and the existing ones. Firstly, we review studies on user identity linkage in multiple online social platforms (OSPs). Although user identity linkage research is out of the dissertation scope, we want to be able to leverage on the solution methods which can be used to link accounts of the same users. Secondly, we examine previous studies on user relationships in multiple OSPs, which focus on: (i) research on structural properties in OSPs, and (ii) link predictions in OSPs. Thirdly, we review existing studies on user activities in multiple OSPs. In particular, we examine the studies on user collaborative activities in OSPs used by the software engineering community. Fourthly, we survey works on modeling user topics in single and multiple OSPs. On the context of multiple OSPs, we also examine works that study user platform preferences. Finally, we review works on identifying influential users in single and multiple OSPs. These include (i) topic oblivious and (ii) topic-specific influential users.

2.1 User Identity Linkage

To study user-generated data across multiple OSPs, we first need to find the OSPs accounts that belong to the same user. The matching of user accounts across multiple OSPS, also known as user identity linkage or network linkage, is a widely studied topic [136, 137, 75, 60, 92, 121, 88, 27]. We can broadly categorize these work into (i) attribute based and (ii) network-based methods. Attribute-based methods derive features from attributes such as emails, names, location, content, usernames, etc., to match user accounts across multiple OSPs [136, 121, 75, 88, 98, 111, 89]. Zafarani and Liu derived over 400 features from usernames and used them for matching users across multiple OSPs [136]. Vasilescu et al., in their empirical study on GitHub and Stack Overflow, utilized email addresses to match users on the two social collaborative platforms [119]. Network-based methods utilize network structures to perform user linkage [143, 82, 77, 90]. Narayanan et al. in a study to analyze user privacy and anonymity in OSPs, proposed a framework to link and identify different accounts of a user using the network structures of users [90]. There are also works that utilized a combination of both attributes and network structures [97, 140, 110, 63]. Kong et al. in particular have proposed to use user attributes, users' ego networks and other spatial, temporal and content information of user accounts in user identity linking[60]. It is important to note that matching user account pairs returned by user identity linkage methods have to be manually examined before they are used as ground truths. In this dissertation, our focus is not on proposing new user identity linkage methods. Instead, we would leverage on these methods to find matching user accounts across multiple OSPs and use the matched user accounts for conducting cross-platform analysis and modeling of user-generated data.

2.2 User Relationships in Online Social Platforms

2.2.1 Structural Properties in online Social Platforms

Since the proliferation of OSPs, there had been a lot of studies that analyzed the structural properties OSPs. Arnaboldi et al. [9], did a study on Twitter and found that a user's social network in Twitter shares similar structural properties with offline social network proposed by Dunbar [33]; i.e., offline social networks are formed by circles of relationships having different social characteristics (e.g., intimacy, contact frequency, and size). Although these works compared users' friendships in OSPs, they are limited to single OSPs. We extend this research to study users' friendships across multiple OSPs.

The study on structural properties and user behaviors in multiple OSPs is an emerging topic gaining the attraction of researchers in recent years. Magnani and Rossi [81] conducted a study on the structural properties in multiple OSPs and proposed to represent multiple OSPs as a *multi-layer network*. They extended the degree and closeness centrality measures to multi-layer networks. Nevertheless, they did not consider other structural properties or behaviors such as friendship similarity and evenness across networks, which will be discussed in Chapter 3 of this dissertation.

A particular structural property which is closely related to our work is the triadic closure property in social platforms. Triadic closure is the property among three nodes A, B, and C, such that if a strong tie exists between A-B and A-C, there will be a weak or strong tie between B-C [113]. The triadic closure property has been widely even before the rise of online social platforms [126]. In recent years, many researchers have studied and attempted to model the process of triadic closure in OSPs. For example, Romero and Kleinberg empirically analyzed the triadic closure process in the Twitter network [104]. Lou et al. performed prediction of reciprocal relationships and triadic closure

process in Twitter. They also developed a model to predict 90% of the reciprocal relationships in Twitter accurately and to predict the links among users [80].

The structural properties in social platforms for the software engineering community have also been studied in recent years [74, 116, 124, 23, 30, 117]. Lima et al. [74] did an extensive macro-level study on the interaction between users in GitHub and found that the number of users involved in repositories follows power-law. Casalnuovo et al. [23] performed an analysis on the role of prior social links on users' collaboration and productivity in GitHub. Thung et al. [116] performed a structural analysis of the user-user and project-project relationships in GitHub and found that software development OSPs are fundamentally different from other OSPs. In software social platforms, users are connected through code while in typical social networks, users are connected directly to each other (i.e., "friends" or "follower" relationship). Similar studies were also conducted in Stack Overflow. Wong et al. [124] in their empirical study on user interactions in Stack Overflow found that most users only answer a few questions and tend to ask and answer questions in similar topics.

Our study in Chapter 3 builds on the existing works and focuses on how similarity and evenness of friendship across social platforms affect the likelihood of triadic closure. On analyzing links in collaborative OSPs, our work in Chapter 4 expands on existing research to investigate the similarity of topical interest among users who are linked by performing collaborative activities together in Stack Overflow and GitHub.

2.2.2 Link Prediction in Online Social Platforms

Link prediction in single OSPs has been a well-studied research problem. Network structural properties are commonly used features to predict links between users [72, 91, 2, 42, 118, 32, 20]. An example would be the neighborhood features, where common neighbors between a pair of users are used to derive some

affinity score which can be used to estimate the likelihood of a link between the two users. [91, 2].

There were also few link prediction studies done on *multidimensional networks*, where two nodes may be connected by more than one dimension, expressing either different types of relationship (e.g., friends, colleagues, relatives), or different quantitative values of the same kind of relationship (e.g., different ranks, or different publication venues for the same co-authorship relation). Rossetti et al. performed supervised and unsupervised multidimensional link predictions on the DBLP and IMDb networks [107]. They proposed to use neighborhood features such as Common Neighbors and Adamic-Adar to predict user collaboration in the different dimensions of a network, e.g., they predicted the collaboration of authors in publishing papers at some venues. Unlike the previous study, we predict friendship of users in different OSPs instead of different dimensions in the same OSP. Multiple OSPs is different from multidimensional networks as only the former requires user identity linkage to be performed. Furthermore, our friendship link prediction methods consider not only friendship neighborhood features but also cross-OSP friendship maintenance features. Our friendship prediction study in Chapter 3 thus investigates beyond structural properties of a user's network in one OSP to cover cross-OSP friendship maintenance features.

2.3 Collaborative Activities in Online Social Platforms

There are many types of user activities in OSPs. For example users may *like* each other posts in Facebook or *retweet* content of other users on Twitter. In this section, we focus on reviewing the existing works on collaborative activities in OSPs used by the software engineering community.

Collaborative activities in OSPs are defined as user activities performed

with the goal of completing a task. For example, in Stack Overflow, users collaboratively *ask* and *answer* software development related questions. Similarly, in GitHub, users can work on repositories together by performing activities such as *watch*, *fork* (i.e., make a copy), *pull-request* (i.e., review codes), *commit codes*, etc.. User collaborative activities in OSPs are widely studied Most of these studies focus on learning the users' topical interests from their collaborative activities, which we will discuss in Section 2.4.1.

There are also few works on analyzing user collaborative activities across multiple OSPs used by the software engineering community. Vasilescu et al. performed a study on users' involvement and productivity in Stack Overflow and GitHub [119]. They found that users who are more active on GitHub (in terms of GitHub commits), tend to ask and answer more questions on Stack Overflow. Badashian et al. [10] conducted an empirical study on the correlation between different types of user activities in the two platforms. Their findings supported the findings of the earlier work by Vasilescu et al., that is: users who actively contributed to GitHub, also actively answered questions in Stack Overflow. They observed an overall weak correlation between the activity metrics of the two networks and concluded that user activities in one network are not strong predictors for activities on another network. Both the works, however, did not consider intrinsic interests of the users, although Vasilescu et al. did mention the possibility of extending their work to consider topic interests of the users. Our work in Chapter 4 fills this gap by examining users' topic interests inferred from the users' collaborative activities across Stack Overflow and GitHub. To our best of knowledge, our work is the first cross-platform study that examines user topical interests in the two OSPs.

Prediction and recommendation of collaborative activities in OSPs have been widely studied. These work can be further categorized into two groups: (i) finding experts to perform a certain platform tasks or collaborative activities [102, 132, 29, 129, 122, 48, 135, 5, 134, 100] and (ii) recommending content

or collaborative activities to users in the OSPs [31, 125, 123, 41, 138, 53]. For work in group (i), there were work which proposed methods to find experts to answer questions in Stack Overflow [102, 132, 29, 129, 122], while for GitHub, experts are predicted if they will review *pull-requests* and software code for repositories [133, 134, 100]. For work in group (ii), Wang et al.[125] conducted a study in Stack Overflow to recommend questions and answers concerning API issues to users. De Souza et al. [31] conducted an experiment to recommend Stack Overflow question-answer pairs relevant to selected software programming problems. Zhang et al. [138] predict and recommend relevant repositories to users based on the users' past collaborative activities (e.g., fork, watch, etc.) in the platform. In a more recent work, Jiang et al. [53] proposed to use user programming language preferences and one-class collaborative filtering to improve prediction of which GitHub repositories are relevant to a user. Our study in Chapter 4 adds on to the state-of-the-art in group (ii) by proposing a novel method that uses user-generated data from multiple OSPs to predict users' collaborative activities in individual OSP.

2.4 User Topics and Platform Preferences in Online Social Platforms

2.4.1 Modeling User Topics in Single Online Social Platform

Topic analysis of OSP users' content is an important research topic for user profiling and recommender systems. Jang et al. proposed to characterize and detect Instagram user age group by applying LDA model [19] to learn the topic interests of teens and adult users [51]. Ferrara et al. conducted an empirical study and analyzed the topic interests of Instagram users using hashtags in the captions of Instagram posts [34].

A similar empirical study was also conducted in Tumblr by Xu et al. using the tags on Tumblr posts [130]. On modeling user interests in Tumblr, Chang et al. applied LDA model on content from Tumblr user posts to discover Tumblr users' latent topic interests [26].

Michelson et al. derived the topic interests of Twitter users by examining the entities mentioned by users in their tweets [85]. Researchers have also proposed to model the topics of tweets and their associated posting activities (e.g., retweet) in Twitter [99, 45]. Hong et al. applied the LDA model and author-topic model [106] to discover the topic interests among Twitter users [46]. Further research works were also done to improve the performance of LDA model by experimenting with different ways of forming documents using tweets [83]. Other works also proposed to model individual user and community topic interests [44] jointly. Our study in Chapter 5 extends these work by jointly learning the user topics across OSPs.

In [141] the researchers proposed TwitterLDA model which is a variant of LDA, in which (a) tweets of the same user are aggregated to form documents; (b) each user has a topic distribution; (c) users share a common background topic; and (d) a topic is assigned to each tweet. It is important to note that TwitterLDA was designed to learn topics from a single OSP, which is different from our proposed model in Chapter 5 where we take into consideration the topic distributions for different OSPs.

There are issues in applying standard topic models, which are designed for single OSP setting, on multiple OSP setting. Suppose we apply the standard topic models on a combined user-generated data from multiple OSPs, the existing models may be able to learn the collective topical interests of a user across multiple OSPs but not the platform-specific topical interests. For example, applying TwitterLDA on combined user-generated data from two OSPs, p_1 and p_2, we may learn that a user u_1 is interested in *music* and *politics*. However, u_1 prefers to discuss her music interests in p_1 while only discuss politics in p_2, and

these platform-specific topical interests are not learned by the existing topic models designed for single OSP setting. Our proposed model in Chapter 5 is designed to address this gap and learn the platform-specific topical interests of users in multiple OSP setting.

The modeling of user topics in OSPs used by the software engineering community was also extensively studied. For example, there are research works that focused on discovering questions topics asked by Stack Overflow users [14, 11, 144, 105]. Similarly, there are also works on mining programming languages used by the users in GitHub [101]. Our work in Chapter 4 extends this field of work by examining the topic interests of users who have accounts on both Stack Overflow and GitHub.

2.4.2 Modeling User Topics in Multiple Social Platforms

There are also works that apply topic models on multiple OSPs. Guo et al. proposed a model that considers social-relationship among users for topic modeling and applied their model on Sina Weibo and Twitter datasets [43]. Cho et al. designed a model that incorporates users' social interactions and attributes for topic modeling and applied their model on six OSPs [28]. However, these works do not link the users across OSPs but perform the topic analysis on each platform independently. Our research differs from such studies by analyzing topical interests of a set of common users with accounts on multiple OSPs.

2.4.3 User Platform Preferences

Despite the increase in cross-OSPs studies, there are relatively few studies on user cross-platform content publishing behaviors. Meo et al. presented a macro-level analysis of users sharing activities on Flickr, Delicious and StumbleUpon [84]. Ottoni et al. studied the users' activities across Twitter and Pinterest and found that users tend to post items to Pinterest before posting them on Twitter [94]. Similar observations were made by Lim et al. who also

found that users exhibited varied information sharing activities on different OSPs[73]. While both [94] and [73] investigate the posting of same content across multiple OSPs, i.e., the duplication of posts across different OSPs, the topic interests and the diverse types of content are however neglected. For instance, a user may not simply duplicate and share a post across OSPs. Instead, she may share different types of content that share the same topic interests across different OSPs. For example, a user may share a text post in Twitter and a photo on Instagram. Although the types of content shared on the two OSPs are different, both the text and photo may share the same topic (e.g., Food). Our study in Chapter 5 attempts to bridge this gap in the state-of-the-art works by examining the topic interest of the diverse types of content published by users on multiple OSPs. Furthermore, we also attempt to study how the topic interests of a post could influence the user's platform choice to publish the post. For example, a user who is interested in architecture design and fashion may choose to share his architecture design posts in Tumblr while sharing the fashion posts on Instagram.

2.5 User Influence in Online Social Platforms

2.5.1 Identifying Topic-Oblivious Influential Users in Online Social Platforms

Many previous works apply network centrality measures to identify topic-oblivious influential users in an OSP [56, 54, 61]. Kayes et al.[56] aggregated network centrality measures such as *degree* [36], *betweenness* [35], *closeness* [36] and *eigenvector* [21] to measure and identify influential bloggers. There are also works which extended HITS algorithm [58] to find influential users in OSPs. Romero et al. [103] proposed the influence-passivity (I-P) algorithm to measure Twitter users' influence and passivity from their retweet activities. Gayo-Avello [37] applied HITS on Twitter follow links to identify and differen-

tiate influential users from spammers. Shahriari and Jalili [109] modified the HITS and PageRank [95] algorithms to analyze and rank users in signed OSPs.

Besides user relationships, user activities, e.g., *retweet* and *mention* in Twitter, can also be used to determine influential users in OSPs. Khrabrov and Cybenko [57] adapted PageRank [95] algorithm to Twitter *mention* activities to identify influential Twitter users. Silva et al. [112] employed a similar approach to find and recommend influential users based on other users' *retweet* activities. Aral and Walker [8] conducted a randomized experiment on Facebook to identify influential and susceptible users based on users' product sharing and adoption activities.

Some studies have also identified influential users by analyzing both user relationships and activities. Agarwal et al. [3] proposed a model that utilizes the page-linking activities to measure the influence of bloggers. Ghosh and Lerman [38] applied centrality measures on Digg users' friendship and voting activities to identify influential users. Cha et al. [25] evaluated the influence of Twitter users using *follower*, *mentions* and *retweets* counts. Other works have also analyzed both user ego networks and tweet activities to find influential users in Twitter [131, 62, 7, 50, 71].

2.5.2 Identifying Topic-Specific Influential Users in Online Social Platforms

Many existing works have the modeling and extraction of topics as the first and separate step in the identification of topic-specific influential users. Commonly, the topics of user-generated content are first determined by performing keyword matching with a topical lexicon [12, 96, 47, 79, 67, 87, 93]. For example, in a study to identify topic-specific authorities in Twitter, Pal and Counts [96] first extracted tweets covering three topics: "oil spill", "world cup" and "iphone" using simple substring matching before applying models to determine the topic-specific authorities from the users' *retweet* activities. Oro et al. [93]

proposed *social media authoritative user (SocialAU)* model which includes a three-layer network (i.e., *user-item-lexicon*) for finding authority and hub users of a pre-defined selected topic by extending the TOPHITS, a model proposed by Kolde et al [59] to analyze a semantic graph that combines anchor text with the hyperlink structure of the web. Instead of pre-defined topics, some studies use topic modeling such as *Latent Dirichlet Allocation* (LDA) [19] in the first step [127, 55, 4, 45, 49, 6]. For example, Weng et al. [127] first applied LDA to learn the latent topics from users' tweets before applying a PageRank-like model called TwitterRank to measure the topic-specific influence of Twitter users. Huang et al. [49] also similarly applied LDA before applying their graph partitioning model to find influential users on Twitter. Hoang and Lim [45] learned the latent topics using Twitter-LDA [141], a model which extends LDA to short-text messages, before analyzing the virality and susceptibility of Twitter users.

There are relatively very few works that jointly model user topical interests and influence altogether. Liu et al. [78] proposed a two-step model which consists of a generative model to learn the direct influence between users and a topic-level influence propagation method to mine the indirect and global influence. In the generative step, the researchers modeled the generation of a user's posts, which is assumed to be either influenced by his or her friends who have the same interests or generated depending on his or her topical interests. Bi et al. [17] introduced FLDA, a Bernoulli-Multinomial mixture model which models the users' topic-specific influence and content-independent popularity. Barbieri et al. [13] proposed the WTFW model, which models topical and social relationships of users. The model learns the authoritative and susceptible users for each topic, and it considers a topic-specific susceptible user to be one who is interested in the topic (e.g., posting topic-related content), and a topic-specific authority user to be one who is followed by many topic-specific susceptible users. In Chapter 6, we extended HITS and proposed a novel

model to jointly models the users' topical interests, hub and authority scores simultaneously. We also propose a multiple platform version of our proposed model, where it can identify topic-specific hubs and authorities across multiple OSPs by jointly learning the users' topical interests, platform preferences, topic-specific hubs and authorities scores from user-generated data in multiple OSPs.

Part I

Empirical Studies

Chapter 3

Analyzing Friendships in Multiple Online Social Platforms

In this chapter [64], we analyze how users maintain friendship in multiple online social platforms (OSPs) by studying users who have accounts in both Twitter and Instagram. Specifically, we measure the similarity of a user's friendship and the evenness of friendship distribution in multiple OSPs. Based upon our empirical study, we conduct link prediction experiments to predict missing friendship links in multiple OSPs using the neighborhood features, neighborhood friendship maintenance features, and cross-platform features.

3.1 Introduction

The participation in multiple OSPs implies that users have to stretch and spread their already limited time and attention over the different OSPs, which results in new dynamics in the maintenance of friendships. For instance, a user may choose to connect to the same group of friends in multiple OSPs for ease of social interaction. On the other hand, another user may partition his friends into groups and connect to different groups of friends in different OSPs except

very few close friends he maintains across multiple OSPs. In this chapter, we propose two friendship maintenance measures, namely, *friendship similarity*, which measures the amount of overlap between a user's friendship networks in multiple OSPs, and *friendship evenness*, which measures the evenness of user's friendship distribution in multiple OSPs. The proposed friendship maintenance measures are applied to empirically analyze how users maintain friendship in multiple OSPs by studying users who have accounts in both Twitter and Instagram. We also address the research question of how one conducts friendship prediction in the context of multiple OSPs. In particular, we explore using the friendship maintenance measures as features to improve the friendship prediction accuracy.

The study on users' friendship maintenance will provide some new insights into other user-generated data studies on multiple OSPs. Lim et al. conducted an empirical study on user's information sharing activities in six OSPs and found users exhibited varied information sharing activities on different OSPs [73]. They postulated that this was due to the difference in user's usage for different OSPs. From friendship maintenance perspective, a possible explanation could be the users were varying their sharing of information to cater for the different groups of "audience" (i.e., friends) in different OSPs. Thus, research on friendship maintenance of users can potentially help to provide new insights to other user-generated data in these OSPs.

The study on friendship in multiple OSPs has real-world applications. In the second part of this chapter, we extend our empirical research on user's friendship maintenance in multiple OSPs and propose friendship maintenance related features to predict missing links (i.e., friendship) in multiple OSPs. There have been few recent link prediction studies done on *multidimensional networks* which refers to networks with multiple types of links between nodes. Some of them applied neighborhood features such as Common Neighbors and Adamic-Adar on a type of links in the network to predict another type of

user's links within the same network [107]. However, it is important to point out that there are differences between multidimensional networks and in multiple OSPs. For example, user accounts need to be matched across OSPs in the multiple OSPs setting, while users account matching is not required in multidimensional networks. In the multiple OSPs setting, user-generated data of a user account in one OSP is only observed by the user's neighbors in the same OSP but not the same user's neighbors in another OSP. On the other hand, in multidimensional networks, user-generated data are observed by all neighbors of the multi-dimensional network. As such, the link prediction in our study is different from that in multidimensional networks.

The study in this chapter is conducted on a large real-world dataset consisting of about 100,000 users on both Twitter and Instagram with tens of millions online friends. This chapter is divided into two main parts addressing different research questions. In the first part, the research question is how users maintain friendship across OSPs. We focus on friendship maintenance measures that allow us to quantify *friendship overlapping* and *friendship distribution*. In the second part of our study, we address the research question of how one conducts friendship prediction in the context of multiple OSPs. In particular, we would like to explore using the friendship maintenance measures as features to improve the friendship prediction accuracy.

As shown in Figure 3.1, our proposed research framework begins with data crawling from both Twitter and Instagram to assemble a dataset of base users. For this set of users, we perform *cross-platform friend matching* to identify the Twitter and Instagram friends of the same users. We then propose several measures to quantify their friendship maintenance. Finally, we use our findings to design both unsupervised and supervised friend prediction methods.

The work in this chapter improves the state-of-the-art of OSP analysis and link prediction in multiple OSPs. We establish a novel research framework to compare friends in two OSPs. Included in the framework are the measures for

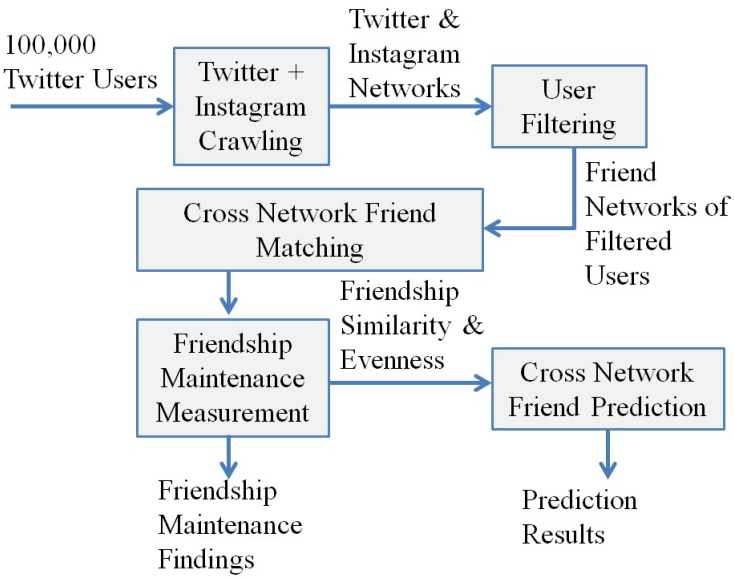


Figure 3.1: Research framework for multiple OSPs friendships analysis

evenness of friendship distribution and similarity of friendship across multiple OSPs, as well as the prediction of links in the multiple OSP setting.

The rest of this chapter is organized as follows: Section 3.2 describes the construction of our Twitter and Instagram datasets. We propose measures that quantify the evenness of user's friendship distribution and similarity of friendship in multiple OSPs in Section 3.3. We then conduct an empirical study in Section 3.4 by applying the proposed measures to analyze the users' friendship maintenance in Twitter and Instagram. Subsequently, we describe the friendship link prediction experiments conducted using friendship features and present the results in Section 3.5. Finally, we conclude this chapter in Section 3.6.

3.2 Data Preparation

To study the user friendships in multiple OSPs, we first need to construct a dataset of users who have accounts with both Twitter and Instagram, a popular microblogging site and a photo-sharing social media site respectively. As the two selected OSPs serve different purposes, it is unlikely that the two OSPs cannibalize each other's users. Furthermore, the two OSPs are highly

complementary and popular among teen users [24]. We, therefore, expect a user on both Twitter and Instagram would generally have the interest to include the same friends in both OSPs.

3.2.1 Dataset

We begin by gathering a set of 100,000 Twitter users who have declared their Instagram accounts in their Twitter biography description from *Followerwonk*[1], a Twitter analytics platform. Subsequently, the Twitter and Instagram followers and followees of these 100,000 users were crawled using the Twitter and Instagram APIs. However, as some of these Twitter and Instagram accounts have set their privacy settings to "private", we are not able to obtain all the followers and followees of the users. We are also only interested in analyzing friendship of average OSPs users; thus we further filter away celebrity or popular users who have more than 2,000 followers. In the end, we manage to obtain 97,978 users who have declared both their Twitter and Instagram accounts, and these users constituted the **base user set**.

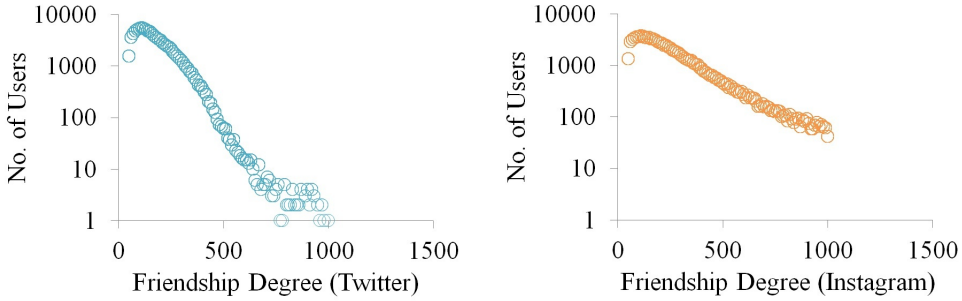


Figure 3.2: Twitter and Instagram friendship distributions

Next, we retrieve the Twitter and Instagram friends of the users in *base user set*. As Twitter and Instagram only capture follower and followee relationships, we define the *friend* of a user to be someone who follows and is followed by the user [128, 52]. In total, we obtained an estimated 17 million Twitter friends and 24 million Instagram friends. Figure 3.2 shows the Twitter and Instagram

[1]https://moz.com/followerwonk/

friendship degree distributions. The average Twitter and Instagram friendship degrees for these users are 171 and 245 respectively.

3.2.2 User Accounts Matching

Before we can study how the users maintain friendships in their Twitter and Instagram accounts, we are required to match the friend accounts in the two OSPs. Unfortunately, very few of the friends have declared both their Twitter and Instagram accounts. Hence, in this section, we present a few simple but effective ways to match users between OSPs by adapting the methods proposed by Zafarani and Liu [136] and Vosecky et al. [121], which are quite effective in this context. We match the Twitter and Instagram friends of our base user set using three levels of user matching methods:

1. **Self-Report Matching**. This method matches the Twitter and Instagram friends of the base user set if these friends declare both their Twitter and Instagram accounts.

2. **Username Matching**. Past research has reported that 59% of users prefer to use the same username repeatedly on different OSNs for easy recall [136]. Instead of matching all our Twitter and Instagram users by their usernames, we match Twitter users with Instagram users by username when they are the friends of the same user in our base set. Doing so minimizes the possibility of two users being matched because they adopt a more popular username.

3. **Username Bigram Matching**. Users may tweak their usernames slightly across different OSNs due to the unavailability of their usual usernames. To cater for such situations, we introduce an approximate method which matches the Twitter and Instagram friends of the base users using username bigrams. Each username is represented by a vector of bigram weights each of which is the number of occurrences of the bi-

gram in the username. Cosine similarity is then applied to two username bigram vectors to determine if the two usernames are sufficiently similar. If the cosine similarity score exceeds a threshold, the two usernames are considered matched. We adopt a threshold value of 0.63 which is derived by taking the median cosine similarity values of Twitter and Instagram username bigrams of the base users.

Table 3.1: Number of users and friends matched using different methods

Methods	Self-Report	Username	Username Bigram	Total
# Users Matched	17,236	1,473,217	1,546,645	3,037,098
# Friends Matched	22,234	1,735,719	1,798,457	3,556,410

Table 3.1 shows the number of friends matched using the above three methods. As expected, the self-report method returns the smallest number of matched friends. A total of 22,234 friends were matched using this method giving an average of $\frac{22,234}{97,978} = 0.23$ matched friends per user. In other words, the vast majority of base users do not have their Twitter and Instagram friends matched using self-report. Username matching method, on the other hand, can match a total of 1,735,719 friends (in addition to those matched by self-report) or an additional 17.72 friends per user, representing $\frac{17.72}{171} = 10.4\%$ and $\frac{17.72}{245} = 7.2\%$ of all Twitter and Instagram friends of the base users respectively. Finally, the username bigram matching method returns yet an additional 1,798,457 matched friends, or 18.36 matched friends per user. This corresponds to 10.7% and 7.5% of all Twitter and Instagram friends respectively. Combining all methods, we can match 3,556,410 friends or 36.3 matched friends per user. Henceforth, we will use all these matched friends in the subsequent analysis.

As there is no ground truth for the validation of the matched friends, we randomly inspected Twitter and Instagram profiles of 100 pairs of matched friend pairs using the username matching and another 100 pairs of matched friends using the combined method. We then looked at the visual cues such

as their profile photos to assess whether the matching methods are accurate. Among the inspected 100 pairs of matched friends using the exact username matching method, we observed that 77 of the pairs have (i) matching profile photos for their Twitter and Instagram accounts, or (ii) their Twitter profile photos matched with some of the photos posted by the Instagram accounts. Majority of the non-matched friend profiles are due to the users not setting profile picture for their Twitter accounts; thus the actual number of matched pairs could be higher than 77. For the username bigram method, 68 of the pairs meet the matching profile photos criteria. This suggests that the user matching methods were able to match the user friends with reasonable accuracy.

3.3 Friendship Maintenance Measures

Before we study how users maintain friendship in Twitter and Instagram, we first propose two measures, *friendship similarity* and *friendship evenness*, to quantify the similarity of user's friendship and the evenness of user's friendship distribution in multiple OSPs respectively.

3.3.1 Friendship Similarity

To ease friendship maintenance, users may choose to overlap their friendships in multiple OSPs. We modified the *D-Correlation* approach by Berlingerio et al. [16] to measure this overlap or similarity of friendship across multiple OSPs. D-Correlation was originally designed for multi-dimensional networks where it measures how redundant are two dimensions for the existence of a node or an edge.

We use $\mathbb{N}$ to denote a set of OSPs $\{N_1, N_2, \cdots, N_n\}$. We denote the set of friends of a user x in a OSP N_i by $FR(x, N_i)$. We define the friendship similarity of user x among these OSPs, $F_{Sim}(x, \mathbb{N})$, to be the ratio of common friends of x across all OSPs as shown in Equation 3.1.

$$F_{sim}(x, \mathbb{N}) = \frac{|\cap_{N_i \in \mathbb{N}} FR(x, N_i)|}{|\cup_{N_i \in \mathbb{N}} FR(x, N_i)|} \tag{3.1}$$

Example. Figure 3.3 illustrates the an example of user distributing his friends in two OSPs, A and B. The user x have a total of 25 friends; 10 friends in A, 20 friends in B, and 5 of the friends are overlapped two OSPs. Thus, the user x's friendship similarity in OSP A and B will be computed as $F_{sim}(x, \mathbb{N}) = 5/25 = 0.2$.

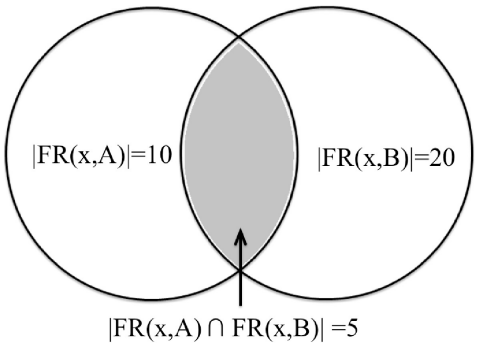


Figure 3.3: Example of user's friendship in two online social platforms

Upper Bound of Friendship Similarity. The maximum Friendship Similarity value (i.e., the upper bound of Friendship Similarity) is only achieved when x has the same friends in all OSPs. The maximum value of a user's friendship similarity in multiple OSPs is equal to the ratio between the minimum and the maximum number of friends added to an OSP among the OSPs that the user has participated (as shown in Equation 3.2). Referencing to the earlier example in Figure 3.3, the maximum possible F_{sim} value for user x would be $10/20 = 0.5$. i.e. user x added all his friends in OSP A in OSP B as well.

$$max(F_{sim}(x, \mathbb{N})) \leq \frac{\min\limits_{N_i \in \mathbb{N}} |FR(x, N_i)|}{\max\limits_{N_i \in \mathbb{N}} |FR(x, N_i)|} \tag{3.2}$$

3.3.2 Friendship Evenness

Suppose that a user x divides all his friends among all the n OSPs without overlap, we expect $\frac{1}{n}$ of his friends in each OSP. Suppose there is a non-zero

overlap among his friends across all the OSPs but negligible overlap between subsets of OSPs, and $F_{sim}(x, \mathbb{N}) > 0$, the *expected ratio of friends x adds to each OSP* is then estimated by $\frac{1}{n} + \frac{F_{sim}(x,\mathbb{N})}{n}$ as shown in Equation 3.3.

$$F_{equal}(x, \mathbb{N}) = \frac{1 + (n-1) \cdot F_{sim}(x, \mathbb{N})}{n} \tag{3.3}$$

Proof. Suppose x has N unique friends in $\mathbb{N}$. Assume that x distributes her friends evenly across the OSPs. Let N_u be the number of unique friends in each OSP and let F denote $F_{sim}(x, \mathbb{N})$. We then expect x to have $N \cdot F$ common friends across the OSPs. In other words, x has $N_u + F \cdot N$ friends in each OSP. As $N = n \cdot N_u + F \cdot N$, we obtain $N = \frac{n \cdot N_u}{1-F}$. Each OSP is then expected to have $N_u + F \cdot \frac{n \cdot N_u}{1-F}$ friends in each OSP. The expected ratio of friends in each OSP is therefore

$$\frac{N_u + F \cdot N}{N} = \frac{N_u + F \cdot \frac{n \cdot N_u}{1-F}}{\frac{n \cdot N_u}{1-F}} = \frac{1 + (n-1) \cdot F}{n} \tag{3.4}$$

When $F = 0$, the above ratio degenerates to $\frac{1}{n}$ implying that all friends of x are equally divided among OSPs exclusively. When $F = 1$, the ratio also becomes 1 suggesting that every OSP covers all friends of x. When there are only two OSPs, i.e., $n = 2$, the expected ratio of friends in each OSP is $\frac{1+F}{2}$.

However, we would expect that in many circumstances, unevenness exists among the friend counts of the OSPs. For example, a user may maintain a larger group of friends in an OSP N_i while keeping a smaller clique in another OSP. We thus define the *ratio of friends of a user x in OSP N_i relative to all friends* in Equation 3.5.

$$F_{in}(x, N_i, \mathbb{N}) = \frac{|FR(x, N_i)|}{|\cup_{N_i \in \mathbb{N}} FR(x, N_i)|} \tag{3.5}$$

Finally, we then define the *evenness of user's friendship distribution* in multiple OSPs as the inverse of summation of the difference between the ratio of friends added in each OSP and the expected ratio of friends a user adds to

each OSP when the friends are evenly distributed as shown in Equation 3.6.

$$F_{even}(x, \mathbb{N}) = 1 - \sum_{i=1}^{n} \Big| F_{in}(x, N_i, \mathbb{N}) - F_{equal}(x, \mathbb{N}) \Big| \tag{3.6}$$

Referring to our earlier example in Figure 3.3, $F_{in}(x, A, \{A, B\})$ is $10/25 = 0.4$ and $F_{in}(x, B, \{A, B\})$ is $20/25 = 0.8$. User x's evenness of friendship distribution in OSP A and B is $F_{even}(x, \{A, B\}) = 1 - (|0.4 - \frac{1+0.2}{2}| + |0.8 - \frac{1+0.2}{2}|) = 0.6$.

Note that $F_{even}(x, \{A, b\})$ measure is also in the range of 0 to 1. Suppose that a user adds an equal number of friends in the two OSPs with any number of overlap friends between the two OSPs, the user's friendship evenness value will be 1. The value for friendship evenness will be 0 if no friend in one of the two OSPs.

There is also an interesting relationship between the upper bound of Friendship Similarity and Friendship Evenness. Based on Equation 3.2, in order to achieve a maximum friendship similarity value of 1 (i.e., $max(F_{sim}(x, \mathbb{N})) = 1$), the minimum and maximum numbers of friends in all the OSPs are identical. That is, user x distributes friendships evenly among all the OSPs ($F_{even}(x, \mathbb{N}) = 1$). Thus, the more evenly distributed the friends among OSPs, the higher the $max(F_{sim}(x, \mathbb{N}))$.

3.4 Empirical Study on Twitter and Instagram

In this section, we apply the friendship similarity and evenness measures to analyze how the 97,978 *base users* maintain their friendships in Twitter and Instagram.

3.4.1 Distribution Analysis

Figure 3.4 shows the distribution of friendship similarity. The average friendship similarity is 0.104. The 1st, 2nd, and 3rd quartile friendship similarity

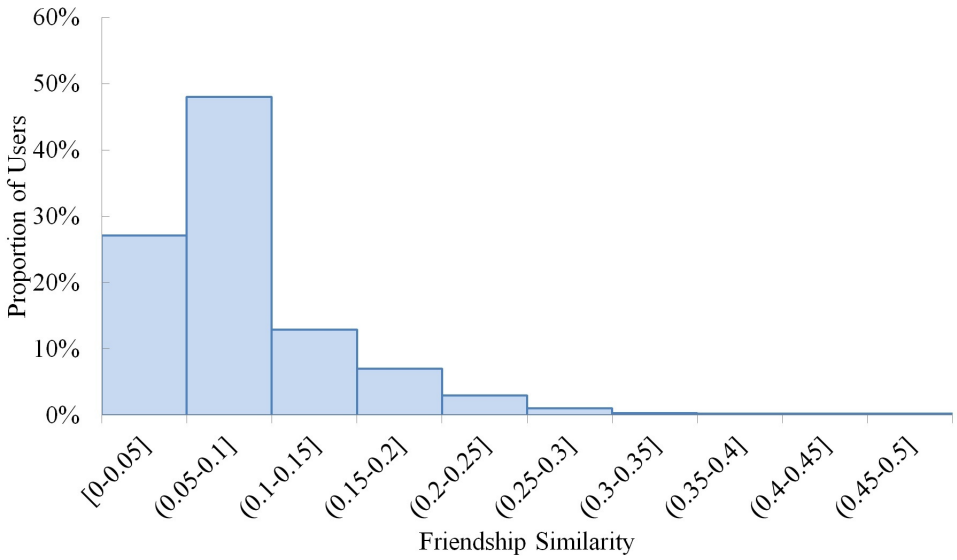


Figure 3.4: Friendship similarity distribution

values are 0.046, 0.09 and 0.148 respectively. This left-leaning bell shape distribution suggests that there are very few users who maintained similar friendship in their Twitter and Instagram accounts. Interestingly, this is contrary to our initial hypothesis that the user would prefer to have a high friendship similarity for ease of maintenance. There could be a few reasons for the low average friendship similarity; for instance, the users may have maintained low evenness for their friendship in the two OSPs, thus limiting the maximum possible friendship similarity value for the users, or the users simply prefer to maintain different groups of friends in different OSPs.

Figure 3.5 depicts the distribution of friendship evenness of the base users. The average friendship evenness is 0.648, a value much higher than the average friendship similarity. The 1st, 2nd and 3rd quartile evenness values are 0.534, 0.705 and 0.856 respectively. The distribution is right-leaning, suggesting that most users may prefer to have not overly uneven friendship counts in different OSPs. Also, the right-learning friendship evenness distribution further strengthens our earlier finding that the users tend to prefer to maintain different groups of friends in different OSPs. There could be many reasons for users' preference to keep the different friendship in different OSPs. One of the possible reasons could be as suggested by Lim et al. [73], that users use

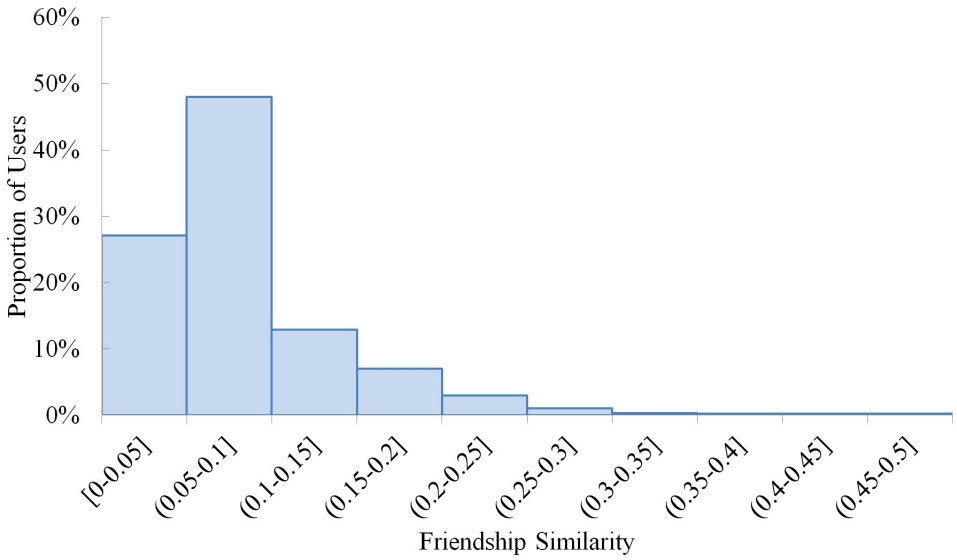


Figure 3.5: Friendship evenness distribution

different OSPs for different purposes or interests, which indirectly motivates the users to connect to different friends in different OSPs. To explain the user's friendship maintenance, we will study beyond the structural properties of multiple OSPs and investigate the differences in the user interests across different OSPs in our subsequence chapters.

3.4.2 Relationship Between Measures

We also examine the relationship between friendship similarity and friendship evenness of users in Figure 3.6 where each point in the figure represents a user with his friendship similarity and evenness values.

Figure 3.6 shows that as the user's friendship evenness increases, friendship similarity seems to increase its range of values. This observation supports what we have highlighted in our earlier discussion that the friendship evenness limits the friendship similarity. We also further investigate this by showing the friendship similarity distribution of users with top and bottom 10% friendship evenness in Figure 3.7. The top 10% friendship evenness users have friendship similarity distribution similar to the overall friendship similarity distribution (as shown in Figure 3.4), while the bottom 10% friendship evenness users have

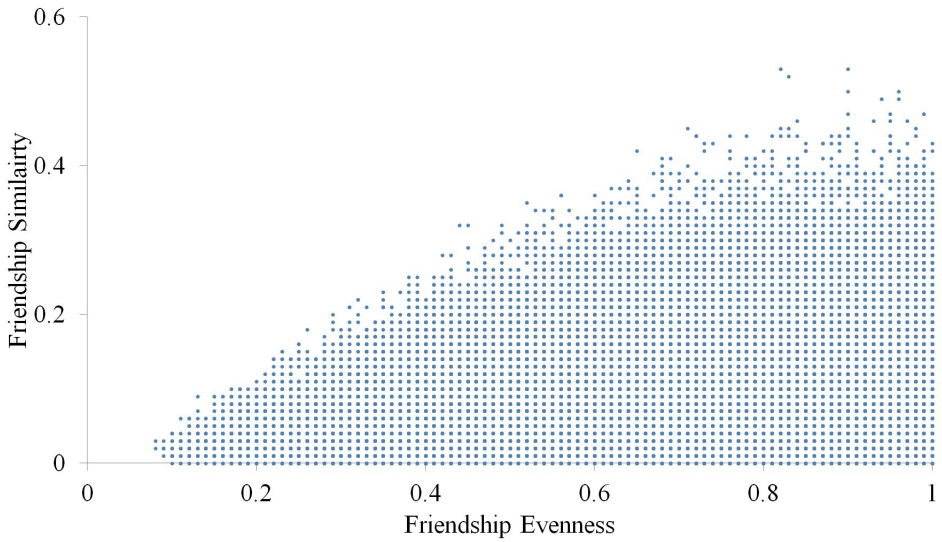


Figure 3.6: Friendship similarity and friendship evenness

a more left-leaning friendship similarity distribution. The top 10% friendship evenness users also have an average of friendship similarity of 0.124, slightly higher than the 0.104 friendship similarity of an average user, while the bottom 10% friendship evenness users have an average of 0.055 friendship similarity, significantly lower than the average user. However, it is observed that there are quite still a number of users who have high friendship evenness but low friendship similarity.

To investigate the dependency of friendship evenness and similarity, we performed a Chi-squared Test of Independence on the two measures. The test result shows p-value $< 2.2e-16$, which is lesser than the 0.05 significance level; therefore we reject the null hypothesis that friendship similarity is independent of friendship evenness. The two measures also show a weak positive correlation of 0.277.

3.5 Friendship Prediction Experiments

In this section, we examine how the link prediction in multiple OSPs can leverage on the links across OSPs. Link prediction can come in two forms; namely, prediction of future links and prediction of missing links [72, 39, 115].

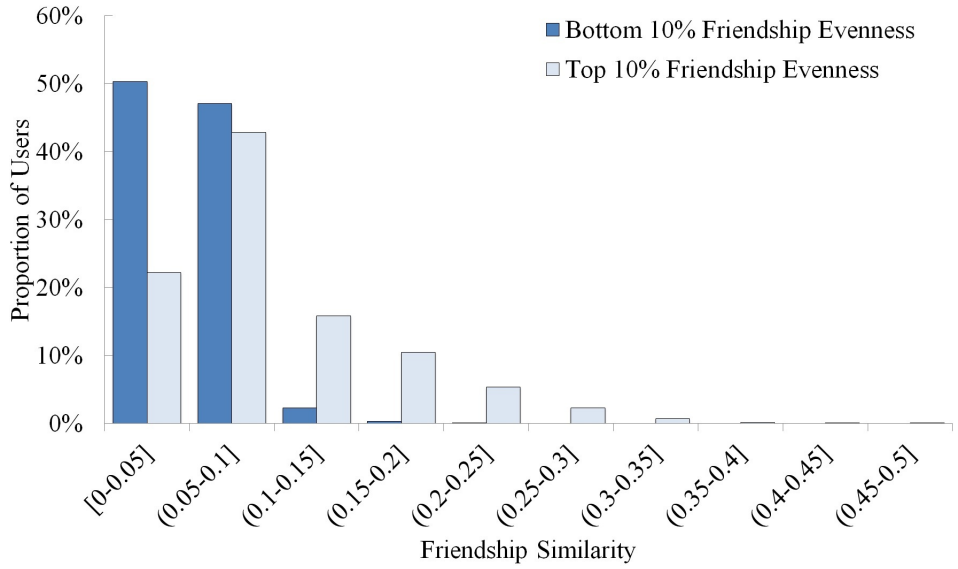


Figure 3.7: Friendship similarity of top and bottom 10% friendship evenness Users

In our research, we focus on the latter which is useful in applications such as friend recommendations. As this is the first attempt to conduct link prediction for multiple OSPs, we also want to answer the following research questions:

- *Can we predict the link between two users in one OSP using the structural information of the two users in another OSP?* Suppose that two users have many common friends in a single OSP, it is likely that they are friends in the other OSP. Intuitively, the existence of a link between the two users in one OSP should also increase the likelihood of a link between the users in another OSP.

- *Can the friendship maintenance features improve the accuracy of link prediction in multiple OSPs?* Now that we have the friendship similarity and evenness measures, we would like to know if they can make good features for link prediction.

3.5.1 Task Definitions

There are two prediction tasks to be performed: (a) **Twitter Link Prediction (TWLP)** where we predict if two users are friends on Twitter; and (b)

Instagram Link Prediction (INLP), where we predict if two users are friends on Instagram.

We now describe the setup of the training and test data in our link prediction task. Let V_{Both} be the 97,978 base users who exist in both Twitter and Instagram. For our base users in Twitter, we define the set of positive instances to be (u, v) pairs such that both u and v are in V_{Both} and (u, v) is an observed link in Twitter. We denote this set of positive instances by $E_{pos}(TWT)$. The set of negative instances, denoted by $E_{neg}(TWT)$, is the set of (u, v) pairs with both u and v from V_{Both} but are not friends in Twitter. The sets of positive and negative instances for our base users in Instagram are defined similarly.

With the above definitions, we derive 17,651 and 26,241 positive instances for base users in Twitter and Instagram respectively, i.e., $|E_{pos}(TWT)| = 17,651$ and $|E_{pos}(INT)| = 26,241$. The numbers look small compared with the size of base users largely because the base users which are selected based on having both Twitter and Instagram accounts do not come from the same user community. Hence, only very few of them know each other on Twitter or Instagram. In other words, there are many more negative instances making the link prediction tasks highly imbalanced. Furthermore, there are additional overheads crawling additional data (e.g., friends or neighbors) for each positive and negative instance in the prediction task. To keep the number of instances manageable for the prediction methods, we randomly select 5,000 positive instances and 25,000 negative instances for each run in our prediction tasks. The negative instances are kept to five times that of positive instances. To make the prediction harder, we also check that at least 5,000 negative instances have at least one common neighbor on Twitter or Instagram.

3.5.2 Unsupervised Link Prediction Methods

We propose to use several unsupervised link prediction methods using different *neighborhood features* as ranking measures[91, 2]. These measures involve using

the common neighbors between a pair of users u and v to derive some affinity score for ranking the user pair. These measures are also based on the triadic closure principle in social network analysis [113]. In this work, the following measures are used:

- **Common Neighbors** (**CN**): This measure counts the number of common neighbors between u and v.
- **Jaccard Coefficient** (**JC**): This measure returns the fraction of common neighbors between u and v.
- **Adamic-Adar** (**AA**): This measure considers the popularity of common neighbors. The less popular common neighbors are given larger weights as they are added together to derive an affinity score.

The above measures are chosen as they were commonly used in link prediction experiments. More formal definitions of them are given at the top of Table 3.2. In Table 3.2, $FR(u, T)$ and $FR(u, I)$ denotes the friends of u in Twitter and Instagram respectively. While applied to score each of the 5,000 positive instances and 25,000 negative instances, the measures are computed using all observable link instances in our dataset, i.e., all links excluding those used as positive instances.

There were also recent studies that applied these neighborhood measures in multidimensional networks, where links between users in one dimension are ranked using the neighborhood features of users in another dimension of the same network [107]. Unlike these existing link prediction works on multidimensional networks, we are now using these neighborhood measures for unsupervised link prediction between users in multiple OSPs where users may not have accounts on both OSPs and users having accounts on both OSPs may not have their accounts matched.

We use *F1 at Top K* to evaluate each unsupervised link prediction method. We first rank all given 30,000 instances by the method's measure in decreasing

order. The *Precision* and *Recall at Top K* are computed by:

$$Prec@K = \frac{\#\text{ correct predictions among top K ranked instances}}{K}$$

$$Rec@K = \frac{\#\text{ correct predictions among top K ranked instances}}{5000}$$

$$F1@K = \frac{2 \cdot Prec@K \cdot Rec@K}{Prec@K + Rec@K}$$

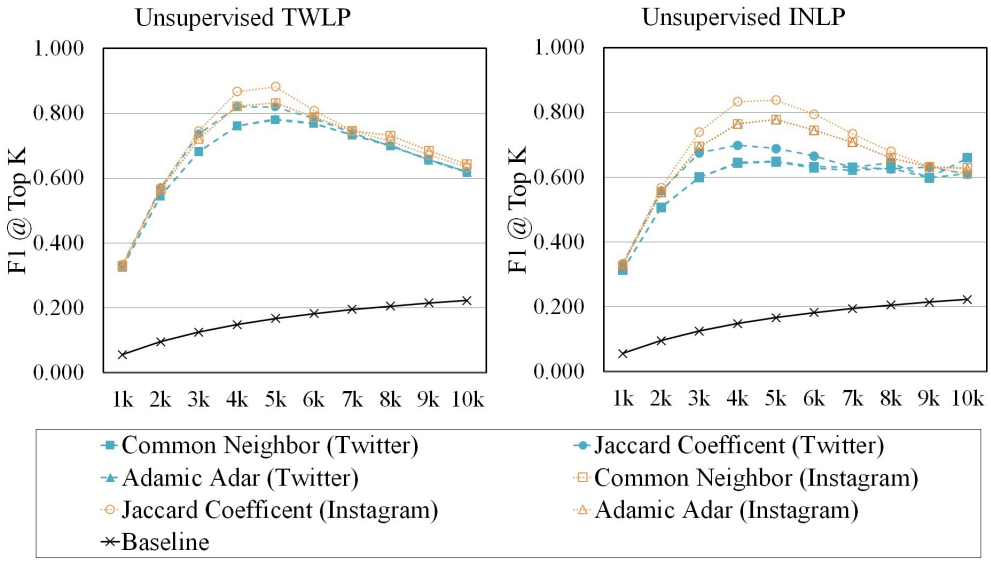


Figure 3.8: F1 scores @ top K for TWLP and INLP

Figure 3.8 shows F1@K of unsupervised link prediction methods in TWLP and INLP tasks. We introduce a baseline method which returns randomly selected K instances as predicted links. We vary K from 1000 to 10,0000 to examine the performance of each method.

As shown in the figure, all the unsupervised methods perform significantly (3 to 4 folds) better than the random baseline in both TWLP and INLP tasks. While the baseline method increases gradually with larger K values due to increasing recall, most of the other methods improve their F1@K only up K=4000 or K=5,000. Beyond which, their F1@K drops. This is because these methods are able to rank positive instances more highly than negative instances.

Interestingly, the figure also shows that the prediction methods using Instagram links outperform those using Twitter links even when the prediction task involves Twitter link prediction, i.e., TWLP. In particular, the method using Jaccard Coefficient on Instagram links (i.e., $\mathbf{JC}_I$) outperforms the rest for almost all K values, achieving the highest F1 scores of 0.882 and 0.838 for TWLP and INLP tasks respectively for top 5,000 ranked results. A possible explanation of the above findings could be that the users have higher friendship degrees in Instagram than Twitter. Two users who are friends on Twitter are likely to have common friends on Instagram. Even though the Twitter neighborhood measures performed worse than Instagram neighborhood measures, they still yield good results (up to 0.689 for F1@5K) in predicting links between users in Instagram. This suggests that predicting links in one OSP using the neighborhood information of another OSP can yield good accuracy.

3.5.3 Supervised Link Prediction Methods

For supervised link prediction, we use Support Vector Machine (SVM) with the linear kernel as the binary classifier trained with each instance represented as a feature vector. SVM is chosen because of its relatively good results in other link prediction tasks. We also consider three types of features as shown in Table 3.2. The **neighborhood features** are the scores from different measures used in unsupervised link prediction methods. By including the neighborhood features, the supervised methods can hopefully achieve at least the good accuracy of the unsupervised methods.

We introduce a binary **cross-platform feature CL** which returns 1 if the users of the instance are friends in another OSP, and 0 otherwise. For example, in the case of TWLP task, a (u, v) instance is assigned a CL feature value of 1 if and only if u and v are friends in Instagram. This feature is included because having a friendship in another OSP should increase the odds of the users having friendship in the target OSP.

Table 3.2: Link prediction features

Feature	**Description**
Neighborhood features	
$\mathbf{CN}_T$	$\lvert FR(u,T) \cap FR(v,T)\rvert$
$\mathbf{JC}_T$	$\frac{\lvert FR(u,T) \cap FR(v,T)\rvert}{\lvert FR(u,T) \cup FR(v,T)\rvert}$
$\mathbf{AA}_T$	$\sum_{z \in FR(u,T) \cap FR(v,T)} \frac{1}{log\lvert FR(z,T)\rvert}$
$\mathbf{CN}_I$	$\lvert FR(u,T) \cap FR(v,I)\rvert$
$\mathbf{JC}_I$	$\frac{\lvert FR(u,I) \cap FR(v,I)\rvert}{\lvert FR(u,I) \cup FR(v,I)\rvert}$
$\mathbf{AA}_I$	$\sum_{z \in FR(u,I) \cap FR(v,I)} \frac{1}{log\lvert FR(z,I)\rvert}$
Common Neighbor Friendship Maintenance features	
$\mathbf{HEHS}_T$	$\frac{\lvert\{z \in FR(u,T) \cap FR(v,T) \lvert F_{sim}(z) \text{ is high}, F_{even}(z) \text{ is high}\}\rvert}{\lvert FR(u,T) \cup FR(v,T)\rvert}$
$\mathbf{HELS}_T$	$\frac{\lvert\{z \in FR(u,T) \cap FR(v,T) \lvert F_{sim}(z) \text{ is low}, F_{even}(z) \text{ is high}\}\rvert}{\lvert FR(u,T) \cup FR(v,T)\rvert}$
$\mathbf{LEHS}_T$	$\frac{\lvert\{z \in FR(u,T) \cap FR(v,T) \lvert F_{sim}(z) \text{ is low}, F_{even}(z) \text{ is low}\}\rvert}{\lvert FR(u,T) \cup FR(v,T)\rvert}$
$\mathbf{LELS}_T$	$\frac{\lvert\{z \in FR(u,T) \cap FR(v,T) \lvert F_{sim}(z) \text{ is high}, F_{even}(z) \text{ is low}\}\rvert}{\lvert FR(u,T) \cup FR(v,T)\rvert}$
$\mathbf{HEHS}_I$	$\frac{\lvert\{z \in FR(u,I) \cap FR(v,I) \lvert F_{sim}(z) \text{ is high}, F_{even}(z) \text{ is high}\}\rvert}{\lvert FR(u,I) \cup FR(v,I)\rvert}$
$\mathbf{HELS}_I$	$\frac{\lvert\{z \in FR(u,I) \cap FR(v,I) \lvert F_{sim}(z) \text{ is low}, F_{even}(z) \text{ is high}\}\rvert}{\lvert FR(u,I) \cup FR(v,I)\rvert}$
$\mathbf{LEHS}_I$	$\frac{\lvert\{z \in FR(u,I) \cap FR(v,I) \lvert F_{sim}(z) \text{ is high}, F_{even}(z) \text{ is low}\}\rvert}{\lvert FR(u,I) \cup FR(v,I)\rvert}$
$\mathbf{LELS}_I$	$\frac{\lvert\{z \in FR(u,I) \cap FR(v,I) \lvert F_{sim}(z) \text{ is high}, F_{even}(z) \text{ is low}\}\rvert}{\lvert FR(u,I) \cup FR(v,I)\rvert}$
Cross-Platform features	
CL	$\begin{cases} 1 & \text{if } (u,v) \text{ exists in another OSP} \\ 0 & \text{otherwise} \end{cases}$

Finally, we also include a group of features known as **common neighbor friendship maintenance features**. While the neighborhood features in one OSP yield reasonable or even good results in unsupervised link prediction in another social OSP, the features may not work very well when the common neighbors demonstrate friendship maintenance that prevents friendship inference across OSPs. For example, a common neighbor between users u and v in Instagram who maintain separate friends in Twitter and Instagram does not increase the likelihood of friendship between u and v in Twitter. The common neighbor friendship maintenance features are obtained by dividing all common neighbors who are present in both Twitter and Instagram into four different categories: namely: (a) high friendship evenness and high friendship similarity; (b) low friendship evenness and high friendship similarity; (c) high friendship evenness and low friendship similarity; and (d) low friendship

evenness and low friendship similarity. We say that a user has high (or low) friendship evenness if her friendship evenness is greater than (or not greater than) the average friendship evenness value. We define the user with high or low friendship similarity in the same way. These common neighbor friendship maintenance features are shown in Table 3.2.

We use six different feature configurations in our supervised link prediction methods as follows:

- **NBO**: Neighborhood features only
- **NFM**: Common Neighbor Friendship Maintenance features only
- **NBOFM**: Neighborhood and Common Neighbor Friendship Maintenance features
- **NBCL**: Neighborhood and Cross-Platform features
- **NFMCL**: Common Neighbor Friendship Maintenance and Cross-Platform features
- **ALL**: All features

We conduct three runs of TWLP and INLP experiments and report the average precision, recall and F1 score of each method. For each run, we use a sample of 5,000 user pairs with friendship and 25,000 user pairs without friendship as the positive and negative instances respectively for training an SVM classifier, and another sample of 5,000 user pairs with friendship and 25,000 user pairs without friendships for testing. We conducted three runs of training and test evaluation altogether.

Table 3.3 shows the results of supervised link prediction for TWLP and INLP tasks. In these experiments, all the feature configurations yield better precision than recall. Most of them have F1 higher than the best F1 scores of the unsupervised methods (i.e., $\mathbf{JC}_I$). Generally, according to F1, the configuration using all features outperforms other methods. Although the Common

Table 3.3: Link prediction results by supervised methods

Tasks	Methods	Avg Prec.	Avg Recall	Avg F1
TWLP	**NBO**	0.954	0.873	0.911
	NFM	0.955	0.830	0.888
	NBOFM	0.953	0.875	0.912
	NBCL	0.976	0.887	0.929
	NFMCL	**0.979**	0.861	0.916
	ALL	0.973	**0.891**	**0.930**
	$\mathbf{JC}_I$	0.882	0.882	0.882
INLP	**NBO**	0.942	0.832	0.883
	NFM	0.959	0.721	0.823
	NBOFM	0.942	0.833	0.884
	NBCL	0.958	0.838	0.894
	NFMCL	**0.971**	0.74	0.84
	ALL	0.956	**0.841**	**0.895**
	$\mathbf{JC}_I$	0.838	0.838	0.838

Neighbor Friendship Maintenance (**NFM**) features performed slightly worse than the Neighborhood (**NBO**) features, the **NFM** features still managed to achieve a reasonably good F1 score of 0.888 and 0.823 for TWLP and INLP tasks respectively. This observation suggests that we can predict, with reasonable accuracy, the friendship between users using the common neighbor's friendship maintenance as features. The addition of Cross-Platform (**CL**) feature also improves the results of **NFM** and **NBO** features. Interestingly, the configuration with Common Neighbor Friendship Maintenance and Cross-Platform features (i.e., **NFMCL**) yield the best precision result in both TWLP and INLP task. This result suggests that the existence of a link between the two users in one OSP increases the likelihood of a link between the users in another OSP.

A possible reason for Common Neighbor Friendship Maintenance (**NFM**) features performing slightly worse than the Neighborhood (**NBO**) features could be due to the lack of common neighbors with friendship maintenance measures who are also base users. Thus we re-examined the supervised link prediction results and determined the accuracy of link prediction for test instances that have at least one common neighbor who is also a base user.

As shown in Table 3.4, our **NFM** features only method outperformed the

Table 3.4: Link prediction results of test instances with at least 1 base user common neighbor

Task	Methods	Avg Prec.	Avg Recall	Avg F1
TWLP	**NBO**	0.948	0.970	0.959
	NFM	**0.971**	**0.994**	**0.982**
INLP	**NBO**	0.938	0.959	0.949
	NFM	**0.976**	**0.999**	**0.987**

method using **NBO** features by precision, recall and F1 score in both TWLP and INLP tasks. This result suggests that there were several occasions where the **NBO** features only method wrongly labeled a positive instance as negative, but **NFM** features correctly label these instances.

Upon further examination of these test instances, we found that although each user pair has very few common neighbors, the common neighbors falls in the *low friendship evenness and high friendship similarity* friendship maintenance category (i.e., LEHS). The users in LEHS connect to more friends in either Twitter or Instagram while keeping a smaller and potentially closer clique of common friends across the two OSPs. Thus, a pair of users with a LEHS common neighbor is more likely to be friends especially when they belong to the smaller clique of friends in one of the OSPs.

3.6 Summary

In this chapter, we studied how users manage and maintain friendships across multiple OSPs. We constructed a base set of about 100,000 users with Twitter and Instagram accounts and studied the friendship of these users in the two OSPs. We introduced friendship similarity to measure the similarity of friendships between two OSPs. A friendship evenness measure was also defined to quantify the degree of balance a user maintains for the number of friendships in different OSPs. We showed that most users prefer to maintain different friendships in different OSPs while keeping only a small clique of common friends across OSPs.

We also investigated link prediction in multiple OSPs using unsupervised and supervised methods. Our experiments have shown that the conventional unsupervised methods using neighborhood features perform well even when we predict links in one OSP using only the network structural properties from another OSP. We have also proposed a set of platform features and applied them to supervised link prediction method. The experiments showed that the supervised methods with suitable feature sets could improve the accuracy over that of unsupervised methods.

Chapter 4

Analyzing Collaborative Activities in Multiple Online Social Platforms

Increasingly, software developers are using a wide array of online social platforms (OSPs) for collaborative work and learning. In this chapter [68, 69], we analyze the users' collaborative activities and topical interests in multiple OSPs in the context of software engineering. We empirically study the topical interests similarities inferred from collaborative activities among users within and across two OSPs: GitHub and Stack Overflow. We also propose a novel mutliple OSPs prediction framework which predicts users' collaborative activities in multiple OSPs using insights and measures derived from our empirical study.

4.1 Introduction

GitHub[1] and *Stack Overflow*[2] are two popular online social platforms (OSPs) among users in software engineering community. GitHub is a collaborative

[1]https://github.com/
[2]http://stackoverflow.com/

software development platform that allows code sharing and version control. Users can participate in various activities in GitHub, for example, users may *fork* (i.e., create a copy of) repositories of other users. Stack Overflow is a community-based website for asking and answering questions relating to programming languages, software engineering, and tools. Although the two OSPs are used for different purposes, users can utilize both platforms for software development. For example, a user who has interests in Java programming language may fork a Java project in GitHub and answer Java programming questions in Stack Overflow.

In this study, we broadly define the topical interests of a user as the programming related topics inferred from the collaborative activities he or she performed on GitHub repositories and Stack Overflow questions. For instance, when a user *answers* Stack Overflow questions tagged with *javascript*, *jquery*, and *angularjs*, we infer that the user is interested in the three topics. Similarly, when a user *forks* repositories in GitHub which description contains keywords such as *javascript* and *ajax*, we assume that the user is interested in the two topics.

The learning of users' topical interests from their collaborative activities could provide new insights on how users utilize the two OSPs for software development. For example, if users share similar topical interests in GitHub and Stack Overflow, the two OSPs may complement each other for software development. Conversely, if the users display differences between their topical interests in GitHub and Stack Overflow, the two OSPs may have been used in a disjoint manner. The social and community-based features in GitHub and Stack Overflow also add new dynamics to the study of user's collaborative activities and topical interests. For instance, users may find themselves sharing similar topical interests with other users who had performed collaborative activities in common repositories or questions together. Thus, it would be interesting to investigate the collaborative activities and topical interests of users

within and across the two OSPs. In particular, we ask the following research questions: Does a user display similar topical interests in his GitHub and Stack Overflow collaborative activities? (**RQ1**), and does a user share similar topical interests with other users who co-participated collaborative activities in GitHub and Stack Overflow? (**RQ2**).

As these collaborative OSPs gain popularity, many research studies have proposed recommender systems to improve the usability of these OSPs. For example, there are works which predict and recommend relevant Stack Overflow questions and answers to aid users in software development [31, 125, 123]. For GitHub, researchers have proposed methods to predict which software repositories are more relevant to a target user [41, 138, 53]. Nevertheless, many of these studies only consider the users' collaborative activities and topical interests in a single platform when predicting and recommending collaborative activities to the users. In this chapter, we aim to utilize the insights and measures derived from our empirical study to predict collaborative activities in multiple OSPs environment.

This work improves the state-of-the-art of cross-platform studies on collaborative activities in multiple OSPs. It gives several key contributions outline below. Firstly, to the best of our knowledge, it is the first research attempt to study similarity of user topical interests across GitHub and Stack Overflow using large datasets. Secondly, we propose several scores to measure the similarity in user topical interests within and across OSPs. The proposed similarity scores are also applied in an empirical study to quantify the similarity in users' topical interests within and across Stack Overflow and GitHub. Thirdly, we extend our empirical findings and conduct prediction experiments to predict users' collaborative activities in the two OSPs using supervised methods with users' topical interests related features derived from insights gathered in the empirical study. Our prediction experiments show that using supervised methods with our proposed features yield good accuracy in predicting user

collaborative activities in multiple OSPs.

The rest of this chapter is organized as follows: Section 4.2 describes the data collection of our GitHub and Stack Overflow datasets. We propose measures that quantify the similarity of user's topical interests within and across multiple OSPs in Section 4.3. We then conduct an empirical study in Section 4.4 by applying the proposed measures to analyze the users' collaborative activities and topical interests within and across GitHub and Stack Overflow. In Section 4.5, we present a novel framework for predicting user collaborative activities within and across multiple OSPs. We also adopt the measures derived from our empirical studies and proposed a set of novel features used in user collaborative activities prediction. The user collaborative activities prediction experiment and results are discussed in Section 4.6. Finally, we conclude this chapter in Section 4.7.

4.2 Data Preparation

There are two main datasets used in our study; we retrieve collaborative activities from October 2013 to March 2015 of about 2.5 million GitHub users and 1 million Stack Overflow users from open-source database dumps[40][3]. Specifically, the below collaborative activities are retrieved from the two platforms:

- *Fork*: Making a copy of a GitHub repository.
- *Watch*: Bookmarking a GitHub repository and receive notifications of activities on the repository.
- *Commit*: Uploading software codes to a GitHub repository.
- *Pull-Request*: Telling other collaborators about changes made to a GitHub repository.
- *Ask*: Asking a Stack Overflow question.

[3]https://archive.org/details/stackexchange

- *Answer*: Answering a Stack Overflow question.
- *Favorite*: Bookmarking a Stack Overflow question.

As this study intends to investigate users' collaborative activities and topical interests across GitHub and Stack Overflow, we further identify users who were using both OSPs. For this work, we used the dataset provided by Badashian et al. [10], where they utilized GitHub users' email addresses and Stack Overflow users' email MD5 hashes to find the intersection between the two datasets. In total, we identify 92,427 users, which forms our *base user* set. Subsequently, we retrieved the collaborative activities participated by the base users. In total, we have extracted 416,171 *Fork*, 2,168,871 *Watch*, 846,862 *Commit*, 386,578 *Pull-Request*, 277,346 *Ask*, 766,315 *Answer* and 427,093 *Favorite* activities. Our subsequent analysis will be based on this group of collaborative activities participated by the base users.

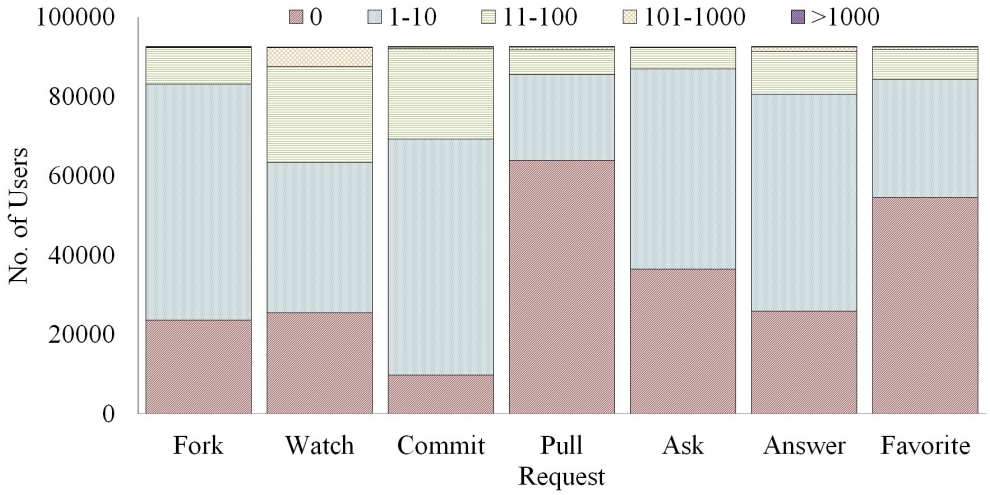


Figure 4.1: GitHub and Stack Overflow base users activities distributions

Figure 4.1 shows the distributions of base users' collaborative activities in GitHub and Stack Overflow. Most of the users forked and committed to 1-10 repositories (64% and 66% of the users respectively), asked and answered 1-10 questions (54% and 59% of the users respectively). There are also quite a number of users who watched 11-100 repositories (26%). We also observe

that more than half of users have at least answered one questions (71%) and a substantial number of users also answered 11-100 questions (12%). Also, more than half of the users had never done a pull-request in GitHub (69%). This is an interesting phenomenon; although both *answer* and *pull-request* could suggest expertise of users, the users seem to engage in more *answer* than *pull-request*. A possible explanation could be the difference in effort required for the two collaborative activities; *pull-request* would require greater effort from the users to learn and edit codes while *answer* would typically require the user to provide a short text answer to a specific question. We also notice that there are users (albeit very few) who were extremely active in GitHub and Stack Overflow; they forked, watched, committed, pull-requested more than 1000 repositories, or asked, answered and favorited more than 1000 questions.

4.3 User Interests Similarity Measures

In this section, we first describe how we infer users' topical interests from their collaborative activities in OSPs. Next, we propose measures to quantify the user's topical interests similarity within and across multiple OSPs.

4.3.1 Inferring User Interests

We infer user's topical interests by observing the repositories and questions that the user forked, watched, committed, pull-requested, asked, answered, or favorited in GitHub and Stack Overflow. We use the following heuristics to infer user's topical interests:

1. To infer user's topical interests in Stack Overflow, we use the descriptive tags of the questions that they asked, answered and favorited. For example, consider a question q related to mobile programming for Android smartphones which contain the following set of descriptive tags: {*Java, Android*}. If a user d asked, answered, or favorited that question, we

infer that his topical interests include *Java* and *Android*.

2. GitHub does not allow users to tag repositories, but it allows users to describe their repositories. These descriptions often contain important keywords that can shed light to user's topical interests. To infer a user's topical interests from the repositories that the user had participated, we first collect all descriptive tags that appear in our Stack Overflow dataset. In total, 39,837 unique descriptive tags are collected. Next, we perform keyword matching between the collected Stack Overflow tags and a GitHub repository description. We consider the matched keywords as the inferred topical interests. We choose to use Stack Overflow tags to ensure that user's topical interests across the two platforms can be mapped to the same vocabulary.

We denote the inferred topical interests of a user given a repository r that he or she forked, watched, committed or pull-requested in GitHub as $I(r)$. Similarly, we denote the inferred topical interests of a user given a question q that he or she asked, answered, or favorited in Stack Overflow as $I(q)$. Since the inferred interests given a repository or a question are the same for all users participated in it, we also refer to $I(r)$ and $I(q)$ as the topics in r and q. For simplicity, we also refer to them as r's topics and q's topics respectively. User d's overall topical interests in GitHub and Stack Overflow, denoted by $I^{GH}(d)$ and $I^{SO}(d)$, is the union of his or her topical interests over all the repositories and questions that d has forked, watched, committed, pull-requested, asked, answered, or favorited.

4.3.2 User Topical Interests Similarity Across Platforms

One way to measure the similarity in an individual user's topical interests across platforms is to take the intersection of his topical interests in Stack Overflow ($I^{SO}(d)$) and his topical interests in GitHub ($I^{GH}(d)$). However, this

simple measure considers all topics to have equal weight. In reality, a user may ask much more questions related to a particular topic than other topics. Similarly, a user may fork repositories associated with a specific topic than other topics. Thus, a finer way to measure the similarity in user's topical interests should consider the number of repositories and questions that belong to each topic.

To capture the above mentioned intuition, we propose *cross-platform similarity score*, which is denoted as $Sim^{SO\text{-}GH}(d)$. Given a user d, we measure d's similarity in topical interests across Stack Overflow (SO) and GitHub (GH) by computing the *proportion* of d's repositories and questions that fall in d's *common topical interests* in Stack Overflow and GitHub (i.e., $I^{SO}(d) \bigcap I^{GH}(d)$). By denoting the repositories and questions that are related to d (i.e., d forked, watched, committed, pull-requested, asked, answered, or favorited these repositories or questions) as $d.R$ and $d.Q$, we can mathematically define $Sim^{SO\text{-}GH}(d)$ as follows:

$$CI(d) = I^{SO}(d) \bigcap I^{GH}(d) \tag{4.1}$$

$$Shared^{Q}(d) = \{q \in d.Q | I(q) \in CI(d)\} \tag{4.2}$$

$$Shared^{R}(d) = \{r \in d.R | I(r) \in CI(d)\} \tag{4.3}$$

$$Sim^{SO\text{-}GH}(d) = \frac{|Shared^{R}(d)| + |Shared^{Q}(d)|}{|d.R| + |d.Q|} \tag{4.4}$$

In Equation 4.1, we define the common topical interests of user d in both Stack Overflow and GitHub. Equation 4.2 defines the set of questions with topics that falls into the common topical interests, while Equation 4.3 defines the set of repositories with topics that falls into the common topical inter-

ests. Equation 4.4 defines $Sim^{SO\text{-}GH}(d)$ as the proportion of repositories and questions of d that falls into the common topical interests.

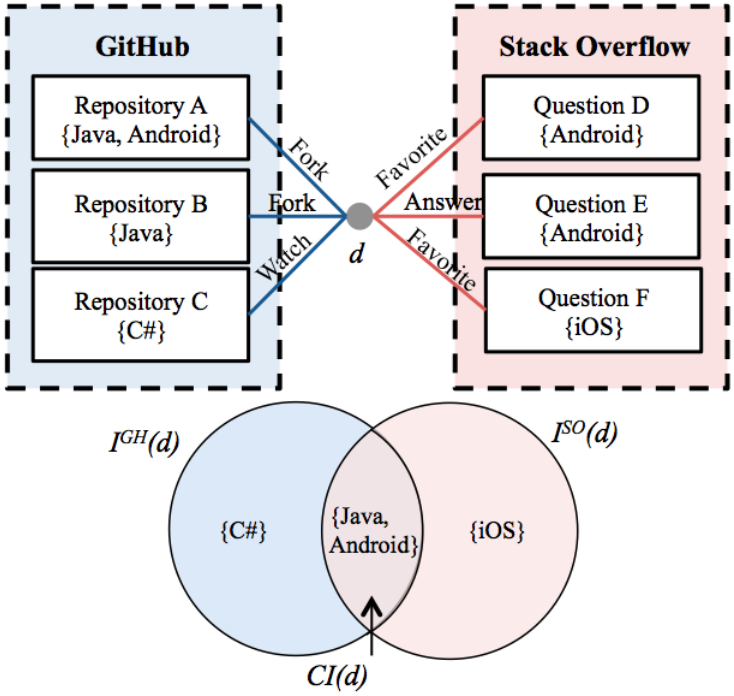


Figure 4.2: Example of *cross-platform similarity score* calculation

Figure 4.2 shows an example for the calculation of *cross-platform similarity score* $Sim^{SO\text{-}GH}(d)$. Consider user d who has performed collaborative activities in GitHub and Stack Overflow. d has forked 2 repositories; *Repository A* which description contains the tag set $\{Java, Android\}$, and *Repository B* which description contains the tag set $\{Java\}$, and watched *Repository C* which description contains the tag set $\{C\#\}$. d also favorited 2 Stack Overflow questions; *Question D* which are tagged with $\{Android\}$, and *Question F* which are tagged with $\{iOS\}$, and answered *Question E* which are tagged with $\{Java\}$. We can estimate d's topical interests in GitHub (i.e. $I^{GH}(d)$) as $\{Java, Android, C\#\}$ and d's topical interests in Stack Overflow (i.e., $I^{SO}(d)$) as $\{Android, iOS\}$. The common topical interests of d (i.e., $CI(d)$) would be $\{Java, Android\}$. Therefore, $Shared^{R}(d)$ would include repositories A and B, while $Shared^{Q}(d)$ would include questions D and E. Thus, $Sim^{SO\text{-}GH}(d) = \frac{|2|+|2)|}{|3|+|3|}$.

4.3.3 User Topical Interests Similarity Among Co-Participating Activity Users

To study the similarity of topical interests among users who performed collaborative activities in GitHub and Stack Overflow together, we propose *co-participation similarity scores*, each focusing on a collaborative activity. Given a collaborative activity and a target user d, we want to measure the similarity between d and *all other users* who co-participated in the target collaborative activity for *at least one* common GitHub repository or StackOverflow question. For example, considering forking a repository as a collaborative activity of interest, we want to find users who co-fork at least one common GitHub repository with d, and we denote the set of other users who co-participated in forking at least one common repository as $Co^F(d)$. Similarly, given a user d, we denote the set of other users who co-participated in watching, committing, pull-requesting, answering, or favoriting at least one common repository or question as $Co^W(d)$, $Co^C(d)$, $Co^P(d)$, $Co^A(d)$, and $Co^V(d)$, respectively.

Intuitively, the more repositories or questions of common topical interests that d share with other users in $Co^F(d)$, the higher the similarities should be. To compute the similarity in topical interests between d and $Co^F(d)$, we measure the average similarity in topical interests between d and each user d' in $Co^F(d)$; for each of such pair, we measure their similarity by computing the proportion of d''s forked repositories which share an interest with the topical interests of d in his or her forked repositories. Mathematically, we define the *co-participation similarity scores* for forking in Equation 4.5.

$$Sim^F(d, Co^F(d)) = \frac{\sum_{d' \in Co^F(d)} \frac{|Shared^F(d,d')|}{|d'.RF|}}{|Co^F(d)|} \tag{4.5}$$

$$Sim^W(d, Co^W(d)) = \frac{\sum_{d' \in Co^W(d)} \frac{|Shared^W(d,d')|}{|d'.RW|}}{|Co^W(d)|} \tag{4.6}$$

$$Sim^C(d, Co^C(d)) = \frac{\sum_{d' \in Co^C(d)} \frac{|Shared^C(d,d')|}{|d'.RC|}}{|Co^C(d)|} \tag{4.7}$$

$$Sim^P(d, Co^P(d)) = \frac{\sum_{d' \in Co^P(d)} \frac{|Shared^P(d,d')|}{|d'.RP|}}{|Co^P(d)|} \tag{4.8}$$

$$Sim^A(d, Co^A(d)) = \frac{\sum_{d' \in Co^A(d)} \frac{|Shared^A(d,d')|}{|d'.QA|}}{|Co^A(d)|} \tag{4.9}$$

$$Sim^V(d, Co^V(d)) = \frac{\sum_{d' \in Co^V(d)} \frac{|Shared^V(d,d')|}{|d'.QV|}}{|Co^V(d)|} \tag{4.10}$$

In the above formulas, $d'.RF$ denotes the repositories or questions that d' forked. Furthermore, $Shared^F\ (d, d')$ denotes the set of repositories which are forked by d' and share common interests with d's forked repositories. Mathematically, it is defined as:

$$\{r' \in d'.RF | \left[I(r') \bigcap \bigcup_{r \in d.RF} I(r) \right] \neq \emptyset\}$$

In Equation 4.5, we define the average similarity in topical interests between user d and other users who had co-forked at least 1 repository with d. The *co-participation similarity scores* for co-watch ($Sim^W(d, Co^W(d))$), co-commit ($Sim^C(d, Co^C(d))$), co-pull-request ($Sim^P(d, Co^P(d))$), co-answer ($Sim^A(d, Co^A(d))$), and co-favorite ($Sim^V(d, Co^V(d))$) are similarly defined in Equation 4.6 to 4.10.

Figure 4.3 shows an example for the calculation of *co-participation similarity score* for watch activity $Sim^W(d, co^W(d))$ for user d. Let us consider two users d and d' and assume that there are no other users. User d watched repositories A and B. User d' co-watched B with d. Thus, $co^W(d)$ is {d'}. In addition to B, user d' also watched repositories C and D. $Shared^W(d, d')$ would then include B and C as both of the repositories share common topical interests with the repositories that d watched. $Sim^W(d) = \left[\sum_{d' \in Co^W(d)} \frac{|2|}{|3|} \right] / |1| = 0.67$.

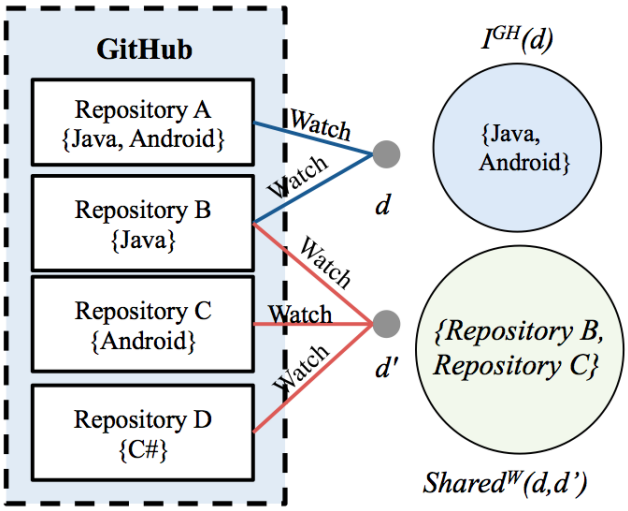


Figure 4.3: Example of *co-participation similarity score* calculation for *watch* activity

It is important to note that the *co-participation similarity scores* only consider the similarity in topical interests between pairs of users who have performed collaborative activities together in at least one common repository or question with each other but the users may have participated in many other repositories and questions different from each other. For example, users d and d' only watched one common repository, but they had watched many other repositories which were different from each other. Also, when computing the co-participation similarity measure between users who participated in a particular activity, we only consider the topical interests of the users in that target activity. For instance, when computing $Sim^{W}(d)$, we consider how similar are the interests between users based only on the *watch* activities, i.e., we do not consider repositories forked by the users or questions answered and favorited by the users.

4.4 Empirical Study on GitHub and Stack Overflow

In this section, we applied the user topical interests similarity measures proposed in the previous section on GitHub and Stack Overflow large datasets. We also attempt to answer the two research questions that we have listed earlier in this empirical study **RQ1** and **RQ2**.

4.4.1 Similarity of User's Topical Interests Across Platforms

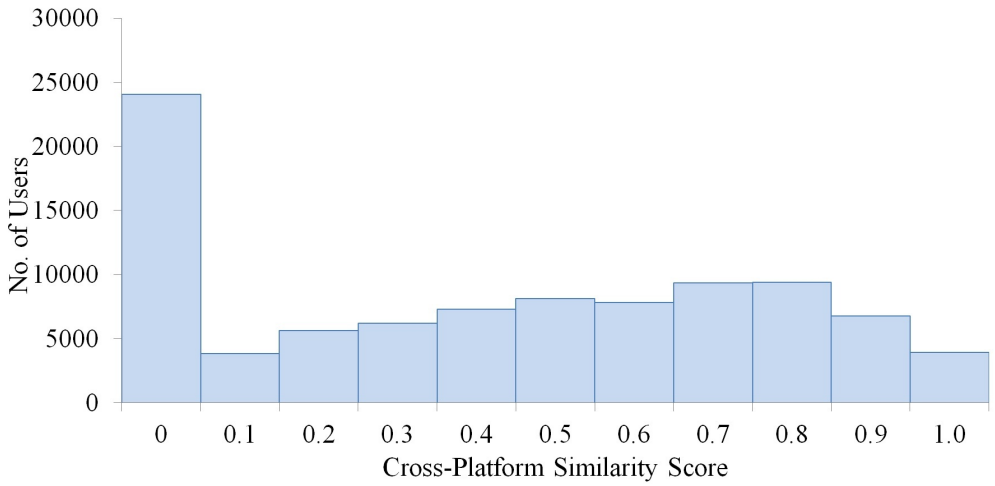


Figure 4.4: Distribution of users' *cross-platform similarity scores* in GitHub and Stack Overflow

Figure 4.4 shows the distribution of the *cross-platform similarity scores* computed for the base users. On average, the users have a similarity score of 0.39. This observation suggests that on average, 39% of the GitHub repositories and Stack Overflow questions that a user had performed collaborative activities on, shared similar topics. Also, close to half (49%) of the users have scored 0.5 or higher, while 26% of the users have their similarity scores equal to 0, i.e., the topical interests of these users are totally different in GitHub and Stack Overflow. This observation suggests that although most users do share

high similarity in topical interests in GitHub and Stack Overflow, however, there are a group of users who have different topical interest in GitHub and Stack Overflow.

We further drill down to compare the similarity for different types of collaborative activity across the two platforms. For example, we measure the similarity in user's topical interests by only considering repositories that the user has forked and questions that the user has answered. 12 different combinations capturing different pairs of collaborative activities across the two platforms are considered: *Fork-Ask*, *Fork-Answer*, *Fork-Favorite*, *Commit-Ask*, *Commit-Answer*, *Commit-Favorite*, *pull-request-Ask*, *pull-request-Answer*, *pull-request-Favorite*, *Watch-Ask*, *Watch-Answer* and *Watch-Favorite*.

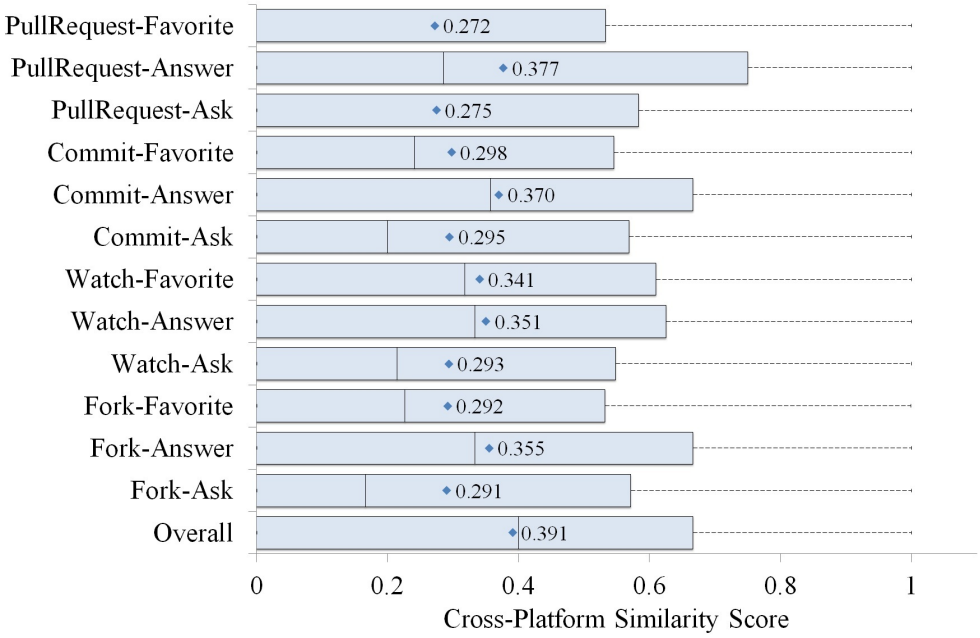


Figure 4.5: Boxplots of users' topical interest similarity for different collaborative activity pairs

Figure 4.5 shows the boxplots of *cross-platform similarity scores* for the 12 different collaborative activity pairs. The collaborative activity pairs have average similarity scores between 0.27 to 0.38, slightly lower than the overall average of 0.39. All the collaborative activity pairs also have a significantly higher number of users with scores of 0. This observation is as expected since by

combining all collaborative activity pairs we have a larger pool of common topical interests. Among the 12 collaborative activity pairs, *pull-request-Answer* pair has the highest average similarity score. A possible explanation for this observation could be attributed to the nature of the collaborative activity; *pull-request* and *answer* not only reveal the topical interests of the users but also demand the users have certain expertise on the topics or programming languages of the participated repositories and questions. For example, a user who is proficient in Java programming language would only *answer* Java programming related questions and submit *pull-request* for Java repositories but he could *watch* other programming language repositories or *favorite* questions from other topics for learning purposes.

4.4.2 Similarity of Interests Among Co-Participating Users

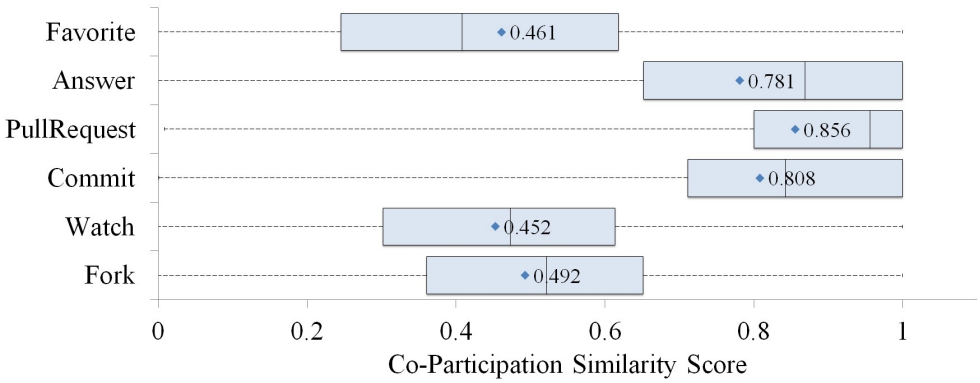


Figure 4.6: Boxplots of *co-participation similarity* scores for different activities

Figure 4.6 shows the boxplots of *co-participation similarity scores* of the base users. We observe that a user has average similarity scores between 0.45 to 0.86 with other users who performed at least one collaborative activity together. This means that given two users who participated in a common collaborative activity, on average 45-86% of all repositories and questions that they participated shared similar topics. Interestingly, we also observed that *commit*, *pull-request* and *answer* have higher average similarity score compare

to the rest of the collaborative activities (0.81, 0.86 and 0.78 respectively). A possible reason for this observation could again be related to the expertise of the users. We would expect that the expertise of the users to be more specialized and less diverse than users' interests, thus resulting in higher similarity scores for users sharing a common *commit*, *pull-request* and *answer*.

4.4.3 Discussion

Our empirical study suggests that users do display some similar topical interests in their GitHub and Stack Overflow collaborative activities (**RQ1**) and users do share common topical interests with other users who have performed collaborative activities together in the OSPs (**RQ2**). Furthermore, we were able to quantify the level of similarity in user's topical interests across different OSPs; we found that on average, 39% of the GitHub repositories and Stack Overflow questions which a user has performed collaborative activities on, share the same topics.

The findings from our empirical study could be extended to build predictive analytics and recommendation application. As we learned that users do share interest similarity across platforms (**RQ1**), intuitively we could predict a user's collaborative activities in one OSP using his or her topical interests displayed on another OSP. For example, if we learn that a user answer Java related questions in Stack Overflow, and he exhibits high similarity in his topical interests across OSPs, we can predict that the user is likely to participate in Java related repositories in GitHub. Similar intuition can be applied to our findings from **RQ2**.

4.5 Activity Prediction in GitHub and Stack Overflow

Figure 4.7 illustrates an example for collaborative activity prediction in a multiple OSP setting. Consider user u, who has accounts on both GitHub and Stack Overflow. If we adopt a *direct platform activity prediction* approach, i.e., predicts a user's activities in a platform using his or her activity interests from the same platform, we could predict that u is likely to answer or favorite question X in Stack Overflow as u has previously answered a $LSTM$ related question. However, if we adopt a *cross-platform activity prediction* approach, i.e., predicts a user's activities in a platform using his or her activity interests from another platform, we could predict that u is also likely to answer or favorite a SVM related question Y as u has previously watched a SVM related repository B in GitHub.

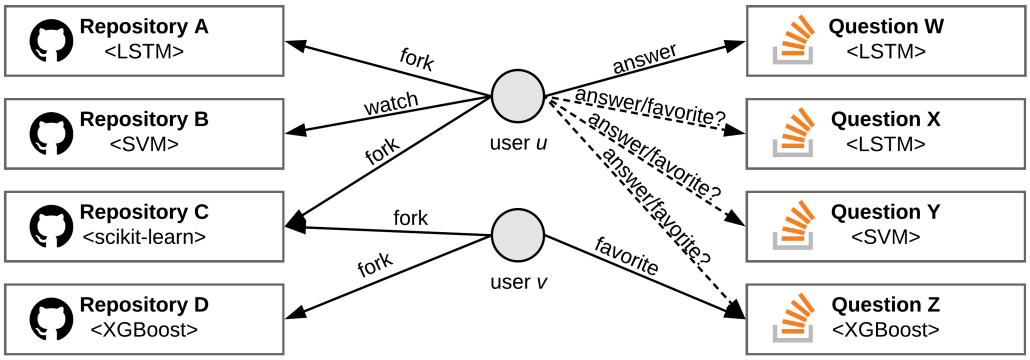


Figure 4.7: Example of Activity Prediction in Multiple Platforms Setting

There are some benefits of using user interests from multiple OSPs for collaborative activity prediction. Firstly, it enables prediction and recommendation of users' collaborative activities in OSPs even when past collaborative activity history of a user is minimal or unavailable, i.e., cold-start problem [108]. For example, if we learn from a user's activities in GitHub that she is interested in *Python* and *text mining* techniques, we would predict that she will likely participate in *Python* and *text mining* related Stack Overflow

questions even when she has just newly joined Stack Overflow and has not participated in any questions. Second, it could cover the *blind spots* of collaborative activity recommender systems which use only data from a single platform. For example, if a user has forked *Android* related repositories in GitHub, recommendation systems which are built on user's past collaborative activity in GitHub will likely to recommend the user more *Android* related repositories. However, the same user may have also participated in some *iOS* related questions in Stack Overflow, and such observations can be used to make relevant GitHub collaborative activity recommendations to the user.

In this section, we first present our proposed multiple platforms collaborative activity prediction framework. We then define the prediction problem and describe the features used in our proposed prediction method.

4.5.1 Multiple Platform Collaborative Activity Prediction Framework

Figure 4.8 shows the framework that we adopt for multiple platforms collaborative activity prediction. We begin with data extraction from two OSPs: Stack Overflow and GitHub. There are three sub-processes in data extraction: (i) matching of users Stack Overflow and GitHub accounts, (ii) extracting the users' collaborative activities, and (iii) inferring users' interests from their activities. The details of these sub-processes are be covered in Section 4.2. Next, we construct the Stack Overflow and GitHub user features which we will use in our collaborative activity prediction.

Our framework also incorporates two approaches to predict users' collaborative activities, namely: *direct* and *cross* platform activity prediction. We define *direct platform activity prediction* as predicting a user's collaborative activity using features from the same OSP. For example, we predict if a given user will answer a given Stack Overflow question using the user's Stack Overflow features. Conversely, we define *cross-platform activity prediction* as predict-

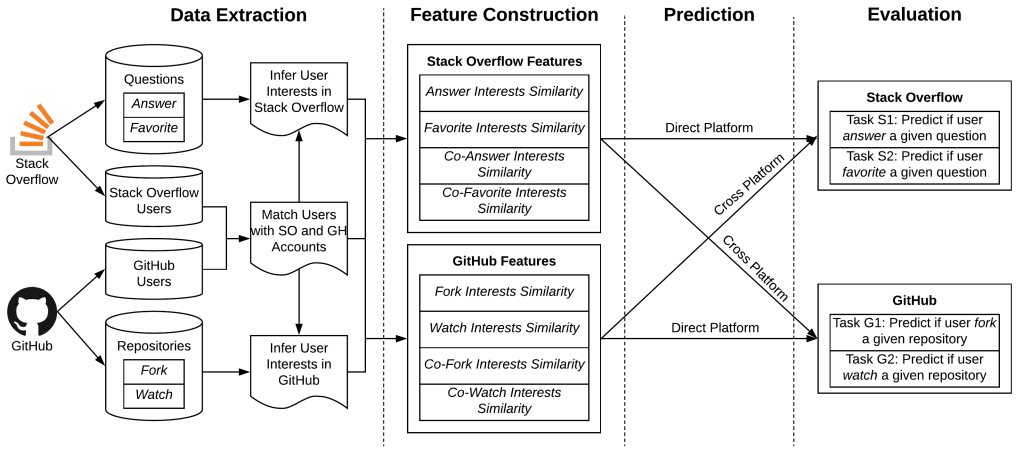


Figure 4.8: Multiple Platforms Activity Prediction Framework

ing a collaborative activity to a user using features from a different OSP. For example, we predict if a given user will answer a given Stack Overflow question using the user's GitHub features. The performance of both prediction approaches will be evaluated on four prediction tasks, which will be described in Section 4.6.

4.5.2 Problem Statement

Given a pair of query user and item (i.e., question or repository), (u, k), we aim to predict if u will perform a collaborative activity (e.g., answer, favorite, fork or watch) on k. There are various ways to measure the likelihood of u performing a collaborative activity on k. For example, we could consider the similarity between k's description and u's topical interests inferred from different activities, or the similarity between k's description and the inferred topical interests of the user who co-participate collaborative activities with u. In our proposed framework, we propose two types of user features, namely:*user activity interest similarity features* and *user co-activity interest similarity features*. The notations used throughout this paper are summarized in Table 4.1.

Table 4.1: List of notations used

Symbol	Description
u	Query user
k	Query item
v	User who co-participated activities with user u
r	Repository
q	Question
$I(r)$	Topics of repository r
$I(q)$	Topics of question q
$I(k)$	Topics of query item k
$u.RF$	Set of repositories forked by user u
$u.RW$	Set of repositories watched by user u
$u.QA$	Set of questions answered by user u
$u.QF$	Set of questions favorited by user u
$Co^{Fork}(u)$	Set of users who co-forked at least one repository with user u
$Co^{Watch}(u)$	Set of users who co-watched at least one repository with user u
$Co^{Ans}(u)$	Set of users who co-answered at least one question with user u
$Co^{Fav}(u)$	Set of users who co-favorited at least one question with user u

4.5.3 User Collaborative Activity Interest Similarity Features

This set of features measures the similarity between a query item k and a query user u's *fork*, *watch*, *answer* and *favorite* collaborative activity topical interests in GitHub and Stack Overflow. The intuition behind this set of features comes from our empirical study, where they found that users in GitHub and Stack Overflow shared similarities between their topical interests in different types of collaborative activities and across the two platforms. Suppose that we want to predict if a user would fork a given repository in GitHub, we would measure the similarity between the given repository's topic and the user's topical interests inferred from the different collaborative activities. Intuitively, the higher the similarity scores, the more likely the user would fork the given repositories. Equation 4.11 captures the above intuition and measures similarity between k and u's fork activity topics (i.e., $Sim_{Fork}(u,k)$), by dividing $\{r \in u.RF | I(r) \in I(k))\}$, which is the number of u's forked repositories that shared common topics with the item interests of k, by the total number of repositories forked by u (i.e.,$u.RF$).

Example. Referencing to the earlier example in Figure 4.7, we could predict if user u will answer question X by computing the similarity between question X and u's fork activity topics. In this example, the common topics between u and question X will be *LSTM*. The number of u's forked repositories that shared common topic with question X (i.e.,$\{r \in u.RF | I(r) \in I(k))\}$) will then be 1 (i.e., Repository A), while the total number of repositories forked by u is 2 (i.e., Repository A and B). Thus, $Sim_{Fork}(u,k) = \frac{1}{2} = 0.5$.

$$Sim_{Fork}(u,k) = \frac{|\{r \in u.RF | I(r) \in I(k))\}|}{|u.RF|} \tag{4.11}$$

$$Sim_{Watch}(u,k) = \frac{|\{r \in u.RW | I(r) \in I(k)\}|}{|u.RW|} \tag{4.12}$$

$$Sim_{Ans}(u,k) = \frac{|\{q \in u.QA | I(q) \in I(k)\}|}{|u.QA|} \tag{4.13}$$

$$Sim_{Fav}(u,k) = \frac{|\{q \in u.QF | I(q) \in I(k)\}|}{|u.QF|} \tag{4.14}$$

We compute the similarities between k and u's watch, answer and favorite activities interests in similar ways as shown in Equation 4.12, 4.13 and 4.14 respectively.

4.5.4 User Co-Participation Interest Similarity Features

This set of features measures the similarity between a query item k and the topical interests of other users v who have performed a collaborative activity together with a query user u. The intuition behind this set of features also comes from our empirical study, where they found that users share similar topical interests with other users who they co-participated a collaborative activity (even minimally) in an OSP. Suppose that we want to predict if a user would fork a given repository in GitHub, we would measure the similarity between

the given repository's topics and the topical interests of other users who had co-forked repositories with the user in GitHub. Intuitively, we would also expect that the higher the similarity score, the more likely the user would fork the given repository. Equation 4.15 captures the above intuition and measures the average similarity between k and fork activity topics of all users v, who had co-forked at least one repository with u (i.e., $Co^{Fork}(u)$).

As users also share common topical interests across different activities and platforms, we would expect that considering other users who had co-participated in other types of collaborative activities with the target user can also potentially help to predict if the target user would participate in a given platform activity. For instance, we are likely able to predict if a user would fork a given repository by measuring the similarity between the given repository's topics and the topical interests of other users who have co-participated with the user in *watch*, *answer* and *favorite* activities.

Example. Referencing to the example in Figure 4.7, we could predict if user u will favorite question Z by computing the similarity between question Z's topic and the topical interests of other users who have co-fork a repository with user u. Assuming that user u only has 1 other user, v, who co-fork repositories with him or her, the common topical interests between v and question Z will be *XGBoost*. The number of v's forked repositories that shared common topics with question Z (i.e.,$\{r \in v.RF | I(r) \in I(k))\}$) will then be 1 (i.e., Repository C), while the total number of repositories forked by v is 2 (i.e., Repository C and D). Finally, $Sim_{CoFork}(u,k) = \frac{\frac{1}{2}}{1} = 0.5$.

$$Sim_{CoFork}(u,k) = \frac{\left[\sum_{v \in Co^{Fork}(u)} \frac{|\{r \in v.RF | I(r) \in I(k)\}|}{|v.RF|}\right]}{|Co^{Fork}(u)|} \tag{4.15}$$

$$Sim_{CoWatch}(u,k) = \frac{\left[\sum_{v \in Co^{Watch}(u)} \frac{|\{r \in v.RW | I(r) \in I(k)\}|}{|v.RW|}\right]}{|Co^{Watch}(u)|} \tag{4.16}$$

$$Sim_{CoAns}(u,k) = \frac{\left[\sum_{v \in Co^{Ans}(u)} \frac{|\{q \in v.QA | I(q) \in I(k)\}|}{|v.QA|}\right]}{|Co^{Ans}(u)|} \tag{4.17}$$

$$Sim_{CoFav}(u,k) = \frac{\left[\sum_{v \in Co^{Fav}(u)} \frac{|\{q \in v.QF | I(q) \in I(k)\}|}{|v.QF|}\right]}{|Co^{Fav}(u)|} \tag{4.18}$$

We compute the similarities between item k's topic and topical interests of other users v who have co-watched, co-answered and co-favorited with a target user u in similar ways as shown in Equation 4.16, 4.17 and 4.18 respectively.

4.6 Collaborative Activity Prediction Experiments

In this section, we describe the supervised prediction experiments conducted to evaluate our proposed method. Specifically, we consider the following Collaborative activity prediction tasks:

- *Answer Prediction.* Given a Stack Overflow *user-question* pair, predict if the user will answer the question
- *Favorite Prediction.* Given a Stack Overflow *user-question*, predict if the user will favorite the question
- *Fork Prediction.* Given a GitHub *user-repository*, predict if the user will fork the repository
- *Watch Prediction.* Given a GitHub *user-repository*, predict if the user will watch the repository

4.6.1 Experiment Setup

Data Selection. For *answer prediction* task, we retrieve all the Stack Overflow questions that the base users have answered and define a positive instance

as a *user-question* pair where a base user had answered the particular question in Stack Overflow. For negative instances, we randomly assign a Stack Overflow question to the base users and check that the randomly assigned pair does not exist in the positive instance set. For the training datasets used in *answer prediction task*, we randomly generated 5,000 negative instances and randomly selected 5,000 positive instances from the questions answered by users between October 2013 and June 2014 (9 months). The same approach was used to generate the positive and negative instances for test sets using the questions answered by the users between July 2014 and March 2015 (9 months). Similar approach was used to generate the *user-question* and *user-repository* pairs for positive and negative instances used in *favorite, fork* and *watch prediction* tasks.

Note that we have repeated the prediction experiments for five runs, and the random selection of train and tests set are repeated for each of the runs. Also, although we know the true labels of the *user-question* and *user-repository* pairs, we do not consider the labels when deriving the values of our proposed features, i.e., we assume that we do not know the labels of the pairs.

Feature Configuration. To compare the performance of *direct* and *cross* platform activity prediction approaches, we use Support Vector Machine (SVM) with linear kernel and apply the following feature sets on all prediction tasks:

- **SO_Act**: This set of features includes the *Answer* (Eqn. 4.13) and *Favorite* (Eqn. 4.14) *Interests Similarity* scores for a given user-question or user-repository pair.
- **SO_CoAct**: This set of features includes the *Co-Answer* (Eqn. 4.17) and *Co-Favorite* (Eqn. 4.18) *Interests Similarity* scores for a given user-question or user-repository pair.
- **GH_Act**: This set of features includes the *Fork* (Eqn. 4.11) and *Watch*

(Eqn. 4.12) *Interests Similarity* scores for a given user-question or user-repository pair.

- **GH_CoAct**: This set of features includes the *Co-Fork* (Eqn. 4.15) and *Co-Watch* (Eqn. 4.16) *Interests Similarity* scores for a given user-question or user-repository pair.
- **ALL**: This set of features is the union of all features.

4.6.2 Prediction Results

We measure the prediction accuracy for each feature configuration by computing the average area under the ROC curve (AUC) over a set of positive and negative examples drawn from the test set for each of the five runs. The results for the four prediction tasks are shown in Figure 4.9. We observe that feature configuration **ALL** performed the best in all prediction tasks, achieving an AUC of 0.89, 0.77, 0.75 and 0.67 for *answer*, *favorite*, *fork* and *watch* prediction tasks respectively.

Performance of cross-platform prediction approach. Although the *cross-platform prediction approach* did not outperform the *direct platform prediction approach* in user collaborative activity prediction, they still yield good accuracy. For example, when predicting user's *answer* and *favorite* activities in Stack Overflow, the GitHub *user collaborative activity interests similarity* features (i.e., **GH_Act**) has AUC of 0.71 and 0.64 respectively, and when predicting user's *fork* and *watch* activities in GitHub, the Stack Overflow *user collaborative activity interests similarity* features (i.e., **SO_Act**) has AUC of 0.65 and 0.58 respectively. The AUC for predicting user's answer activities in Stack Overflow using *user collaborative activity interests similarity* features (i.e., **GH_Act**) is observed to be slightly higher than the prediction for other activities. A possible explanation for this could be the difference between the nature of user activities; answering a question in Stack Overflow would

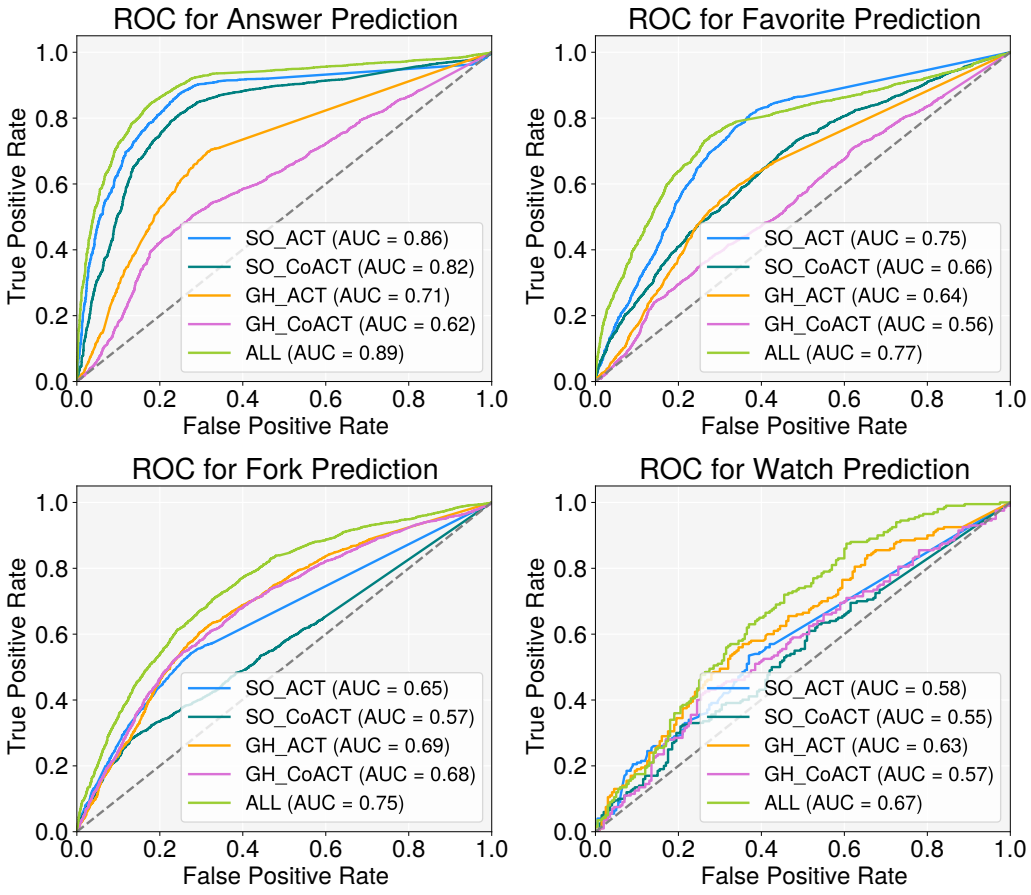


Figure 4.9: ROCs for Four Prediction Tasks

require that a user possesses particular domain expertise, whereas other activities such as watching a GitHub repository or favoriting a Stack Overflow question depend on the user's interests. As such, we observe higher AUC score for *predicting answer activity* task as the users' expertise are usually more specialized and less diverse than their interests.

More interestingly, using *cross-platform prediction approach* with *user co-participation interests similarity* features (i.e., **GH_CoAct** and **SO_CoAct**), have also yielded reasonable prediction accuracy. For example, when predicting user's answer activities in Stack Overflow, **GH_CoAct** has yielded an AUC of 0.62. This result suggests that even with no information about a user's past collaborative activities in the Stack Overflow and only minimal information such as the user's co-participation collaborative activities in GitHub, we are still able to predict user's activity in Stack Overflow reasonably. Similar

observations are made when predicting user activities in GitHub using user's co-activities in Stack Overflow.

4.6.3 Discussion

The results of the four prediction tasks offer us some insights in performing recommendations in OSPs. The reasonably good accuracy of cross-platform prediction approach also demonstrate its potential to solve the cold-start problem; i.e., predicting and recommending a user's activities without knowing the users' past collaborative activity history on the platform. For example, when predicting user's answer activities in Stack Overflow, we can achieve AUC as high as 0.71 without using any Stack Overflow features (i.e., using GitHub features **GH_Act** only). Similar observations were made for the fork, watch and favorite activities.

We further conduct a small case study to retrieve and review fork predictions of users who did not have any past fork activities. For example, we successfully predicted that user *U420338* would forked repository *R12172473* in GitHub even when this was the first repository forked by the user (i.e., no past user fork activity). Examining into details, we found that *R12172473* has description tags $\langle svg, javascript \rangle$, and among the 95 questions *U420338* had answered in Stack Overflow, 83 contain the tags $\langle javascript \rangle$ or $\langle svg \rangle$ or both. By analyzing *U420338*'s Stack Overflow activities, our approach can identify his interests, which ultimately help in predicting the user's GitHub activities.

4.7 Summary

In this chapter, we have studied the similarity in user collaborative activities and topical interests within and across GitHub and Stack Overflow. Our findings were based on data for 92,427 users who were active in GitHub and Stack Overflow. We first proposed similarity scores to measure similarity in

users' topical interests within and across OSPs. Next, we applied our proposed similarity scores in an empirical study on GitHub and Stack Overflow. We observed that on average, 39% of the GitHub repositories and Stack Overflow questions that a user had performed collaborative activities on, shared similar topics. The users also do share common interests with other users who co-participated collaborative activities in the platforms. We also propose a novel framework which predicts users' collaborative activities in multiple OSPs. Our experiments on large real-world datasets have shown that users' collaborative activities in Stack Overflow can be predicted with reasonable accuracy using the same user's topical interests inferred from his or her collaborative activities in GitHub. The same observation was made when predicting a user's collaborative activities in GitHub using his or her topical interests inferred from his or her activities in Stack Overflow. The reasonable accuracy yield by cross-platform prediction approach demonstrates its potential in solving the cold-start problem in user collaborative activity prediction and recommendation in OSPs.

Part II

User Modeling Tasks

Chapter 5

Modeling User Topical Interests and Platform Preferences

With multiple OSPs designed for different purposes and communities, users typically show preferences of certain OSP(s) over others for specific topics. Such platform preferences may even be found at the individual level. In this chapter [65], we model topics as well as platform preferences of users by proposing a new topic model known as MultiPlatform-LDA (MultiLDA). Instead of just merging all posts from different OSPs into a single text collection, MultiLDA keeps one text collection for each OSP but ensures that all OSPs share a common set of topics. MultiLDA further learns the user-specific platform preferences for each topic. We evaluate MultiLDA against TwitterLDA, the state-of-the-art method for OSP content modeling, on two aspects: (i) the effectiveness in modeling topics across OSPs, and (ii) the ability to predict platform choices for each post.

5.1 Introduction

The surge of users using multiple OSPs has opened up new challenges to learning users' topical interests. Learning user topical interests in OSP is a widely studied research topic [34, 26, 130, 85, 46, 141]. Most works study topics in

the text content of OSP. There are also studies that learn latent topics (or clusters) from user behaviors (e.g., forwarding posts, expressing "likes", etc.) and network features [99, 44]. Most of them demonstrated the applications of the learned user topical interests in e-commerce and services recommendation [139, 142]. Nevertheless, all these studies have been confined to textual content from single OSP.

With the same users using multiple OSPs, the holistic approach is to learn user topical interests considering the combined user-generated data from multiple OSPs. For example, one could learn from a user's Twitter data that she is interested in IT gadgets, but the same user is interested in food and fashion based on her Instagram posts. This approach, however, requires two significant challenges to be tackled, namely *user linkage* and *multiple OSPs topic modeling*. The former refers to linking user accounts from different OSPs belonging to the same users. The latter is topic modeling in the multiple OSPs context where heterogeneous media types and users' platform preferences are the additional model elements. User linkage is a highly active research topic but is not the focus of this chapter [121, 136, 137, 27]. In this chapter, we assume that user linkage has already been performed and focus on the second major challenge, multi OSP topic modeling.

As part of this work's research objectives and contributions, we propose a generative model that can learn topics from user-generated data from multiple OSPs as well as their platform preferences. A simple way to perform multiple OSPs topic modeling is to apply an existing topic model such as LDA [19] on the directly combined content of the same users. Unfortunately, such an approach does not work when the content is of different media types, nor does it consider the platform preferences of the users when the latter share content of different topics.

Figure 5.1 shows the methodology used in our research. We first construct a topic model for multiple OSPs. In this work, we propose *MultiPlatform-LDA*

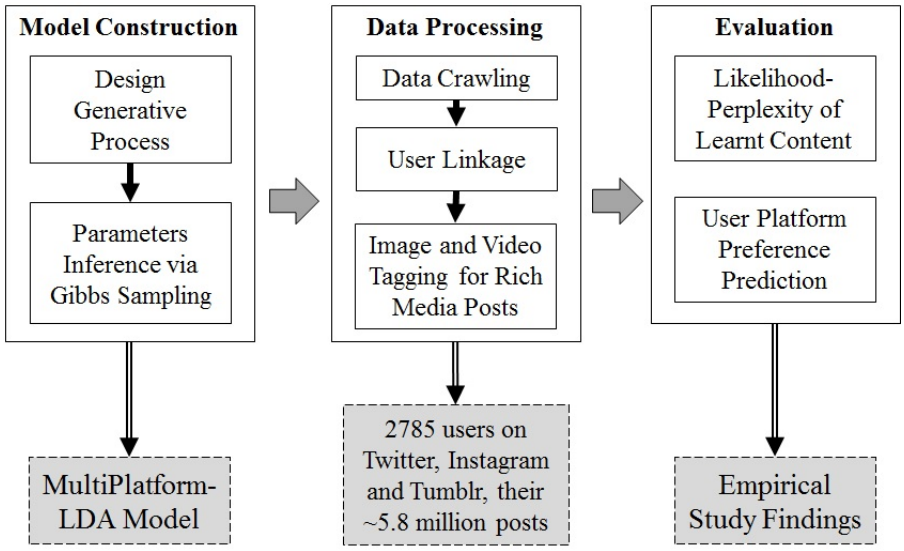


Figure 5.1: Research framework for analyzing user topic-specific OSP preferences

(MultiLDA), a topic model that jointly learns the topical interests and platform preferences of users who have accounts on multiple OSPs. Next, we have a data processing step to gather user-generated data from multiple OSPs, to conduct user linkage (if required) and to turn all rich media content (i.e., images and videos) to words using the state-of-the-art image captioning software. The identification and crawling of this dataset itself is a major challenge. In total, we have gathered about 5.8 million text and rich media posts from 2,785 users who have accounts on Twitter, Instagram, and Tumblr.

Finally, we evaluate the multiple OSPs topic model(s). We perform two sets of experiments to assess MultiLDA: (i) we first use likelihood and perplexity to evaluate the model's ability to learn users' topical interests from the observed text and rich media posts, and (ii) we also evaluate the predictive power of MultiLDA model. Lastly, we also conduct an empirical study on the real-world data using our model, where we learn and report the popular topics on different OSPs and the individuals' platform preferences.

On the whole, this work improves the state-of-the-art topic modeling research and derives several interesting findings. These include:

- In modeling text and rich media content from multiple OSPs, Multi-

LDA outperforms TwitterLDA, another state-of-the-art topic model for modeling social media text.

- In the prediction of users' platform choices, MultiLDA predicted users' platform choice with a high average accuracy of 0.947, outperforming TwitterLDA's average accuracy by 30%.

- In our empirical study, we found different OSPs having different popular topics. E.g., users prefer to post music related topics on Tumblr while sharing food-related topics on Instagram. Also, while most users tend to conform to the general topic distribution of OSPs (i.e., post content with popular topics in the platform), individual user platform preference still exists. MultiLDA was able to model this individual user platform preference effectively.

The rest of this chapter is organized as follows: Section 5.2 describes the construction of our Twitter, Instagram, and Tumblr datasets. We present the MultiLDA model in section 5.3. Section 5.4 presents the experimental evaluations for our proposed model using the real-world datasets. The empirical analysis on users' platform choices and topics on the studied OSPs will also be discussed in Section 5.5. Finally, we conclude this chapter in Section 5.6.

5.2 Data Preparation

Our model evaluation requires a dataset combining user-generated data from multiple OSPs, and we want these OSPs to share some common users. We selected three popular OSPs, namely (a) Twitter, a short-text microblogging site; (b) Instagram, a photo-sharing social media site; and (c) Tumblr, a social networking and blogging site that supports a wide range of rich media such as pictures, videos, etc.

We began by gathering a set of 234,289 Singapore-based Twitter users who declared Singapore location in their user profiles. These users were identified

by an iterative snowball sampling process starting from a small seed set of well known Singapore Twitter users followed by traversing the follow links to other Singapore Twitter users until the sampling iteration did not get many more new users. From these Twitter users, we obtained a subset of them having a user account(s) on Instagram, Tumblr, or both.

Among the above Twitter users, we selected users who also mentioned their Instagram and/or Tumblr accounts (in the form of username or hyperlink) in their Twitter bio descriptions. As some users chose to mention their other OSP accounts on Instagram or Tumblr, we also gathered the linked user accounts of other OSPs by scanning the bio descriptions of Instagram and Tumblr users. As some of these linked user accounts may no longer exist, we performed checking of account existence using the respective OSP APIs. Those user accounts which no longer exist were removed from our dataset. We further filtered away inactive users who did not make at least five posts in the year 2015 on any OSPs.

Table 5.1: Number of users in each particular OSP who use another OSP

	Twitter	Instagram	Tumblr
Twitter	**2696**	2446	272
Instagram	-	**2537**	111
Tumblr	-	-	**362**

In total, we have gathered 2,785 users who form the *base user set*. Table 5.1 shows the breakdown of overlapping users between the three OSPs. Twitter users form the largest group with 2,696 of them (see the first diagonal entry) in the base user set. Instagram is slightly smaller with 2,537 users. Tumblr users form the smallest user group with 362 users. There are 2,446 overlapping users between in our Twitter and Instagram data. The common users between Tumblr and the other OSPs are much fewer. Not shown in Table 5.1, our dataset also has 22 users active on all the three OSPs. Note that this dataset construction is biased towards Twitter which was conveniently used as the first OSP to find the other linked accounts from Instagram and Tumblr. This

bias should not affect our findings if the Instagram and Tumblr users without Twitter accounts have topical interests similar to those with Twitter accounts.

Table 5.2: Number and types of base users' posts in each OSP

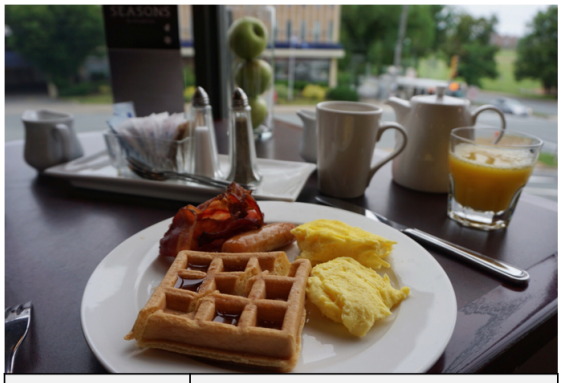
	Twitter	**Instagram**	**Tumblr**
Text	4,923,083	-	135,853
Photo	-	223,325	515,530
Video	-	-	27,015

To learn the users' topics and platform preferences, we gathered all posts generated by each user of our base user set in the year 2015 using the platform-specific APIs. Table 5.2 shows the number and types of posts published by the base users in the three OSPs. From Twitter, we collected nearly 5 million tweets. From Instagram, we gathered 223,325 photo images. From Tumblr, we obtained 135,853 text messages, 515,530 photo images, and 27,015 videos. In total, we have 5.8 million posts from all these base set users to be used in our multiple OSPs topic modeling experiments.

Other than tweets from Twitter and text posts from Tumblr, the photos and videos from Instagram and Tumblr rich media objects have to be converted to text content before we can apply topic modeling on them. One possible way is to extract the user annotated text associated with these photos and videos. Unfortunately, we found that about 23% of our Tumblr posts do not have user annotated text. We also found that the user-provided annotations may not accurately describe the content. In this work, we, therefore, relied on *Clarifai* [1], a third-party visual recognition API that is well known to accurately recognize objects and scenes in rich media, to generate word tags for the photos and videos. The generated tags will then replace the photos and videos in topic modeling. In the case of Tumblr, we thus have posts that are originally text messages as well as posts that are a bag of tags returned by Clarifai.

For example, Figure 5.2 shows a photo posted in Instagram with caption and the *Clarifai* generated tags. While the caption expresses the user opinion

[1] https://clarifai.com/



User Caption:	*Tastes so good!*
Clarifai Tags:	*<breakfast>, <food>, <coffee>, <indoors>, <dawn> <tea>, <bread>, <delicious>*

Figure 5.2: Example of photo posted with caption and Clarifai generated tags

about the food in the scene, the visual recognition tool can better describe most if not all objects in the photo. This makes the generated tags suitable for modeling topics relevant to the photo.

5.3 Modeling Platform Choice and Post

In this section, we present our proposed model, MultiPlatform-LDA (Multi-LDA), which learns the topics and topic-special platform preferences of each user in multiple OSPs.

5.3.1 Notations

Before we present our proposed model, we first summarize the notations in Table 5.3. Given a set of users and their posts on some OSPs, we use $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{P}$ to denote the sets of users, posts, and OSPs respectively. We use S_u to denote the number of users u's posts across all the OSPs. The s-th post of user u is then denoted by the pair $(p_{u,s}, N_{u,s})$ where $p_{u,s}$ is the platform of the post, and $N_{u,s}$ is the content of the post. In this work, we focus on text content and assume that $N_{u,s}$ is a bag of words. The n-th word of the post $(p_{u,s}, N_{u,s})$ is then denoted by $N_{u,s,n}$. Lastly, we use $\mathcal{V}$ to denote the vocabulary of all the

words found in the dataset.

Table 5.3: Notations

Symbol	Description
$\mathcal{V}$	Vocabulary of words in users' content
$\mathcal{U}/\mathcal{S}/\mathcal{P}$	Sets of users, posts and OSPs
K	Number of topics
$\mathcal{S}_u$	Set of posts of user u
$N_{u,s}$	Set of words of s-th post of user u
$p_{u,s}$	Platform of s-th post of user u
$w_{u,s,n}$	n-th word of s-th post of user u
$z_{u,s}$	Topic of s-th post of user u
$y_{u,s,n}$	Coin of n-th word of s-th post of user u
ϕ_k	Word distribution of topic k
ϕ^B	Word distribution of background topic
π	Bias toward background topic
θ_u	Topic distribution of user u
$\sigma_{u,k}$	Platform distribution of user u for topic k
$\mathcal{P}$	Bag of platforms of all posts
$\mathcal{C}$	Bag of coins of all words
$\mathcal{C}_{-u,s,n}$	Bag of coins of all words except $\omega_{u,s,n}$
$\mathcal{Z}$	Bag of topics of all posts
$\mathcal{Z}_{-u,s}$	Bag of topics of all posts except the s-th post of user u
$\mathbf{D}^c_{-u,s,n}$	Tuple $(\mathcal{C}_{-u,s,n}, \mathcal{Z}, \mathcal{S}, \mathcal{P}, \alpha, \beta, \mu, \gamma)$
$\mathbf{n_y}(c, \mathbf{D}^c_{-u,s,n})$	#times in $\mathbf{D}^c_{-u,s,n}$ that words are associated with the coin c
$\mathbf{n_b}(\omega, \mathbf{D}^c_{-u,s,n})$	#times in $\mathbf{D}^c_{-u,s,n}$ that the word ω is associated with the background topic
$\mathbf{n_w}(\omega, z, \mathbf{D}^c_{-u,s,n})$	#times in $\mathbf{D}^c_{-u,s,n}$ that the word ω is associated with topic z
$\mathbf{D}^z_{-u,s}$	Tuple $(\mathcal{Z}_{-u,s}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \alpha, \beta, \mu, \gamma)$
$\mathbf{n_{wz}}(\omega, z, \mathbf{D}^z_{-u,s,n})$	#times in $\mathbf{D}^z_{-u,s}$ that word ω is associated with topic z
$\mathbf{n_p}(p, z, \mathbf{D}^z_{-u,s})$	#times in $\mathbf{D}^z_{-u,s}$ that posts about topic z are associated with platform p
$\mathbf{n_z}(k, u, \mathbf{D}^z_{-u,s})$	#times in $\mathbf{D}^z_{-u,s}$ that posts of user u are associated with topic k

5.3.2 Generative Process

Our model is designed based on the assumption that users have OSPs preference specific to topics. That is, given a topic, users may prefer to generate content about the topic more on a specific OSP than other OSPs. For example, a user may post more gourmet related photos on Instagram but post more tweets about sports and entertainment on Twitter. Thus, to model the users' interests accurately, it is important to learn both the topics of the user-

generated content and topic-specific platform preference.

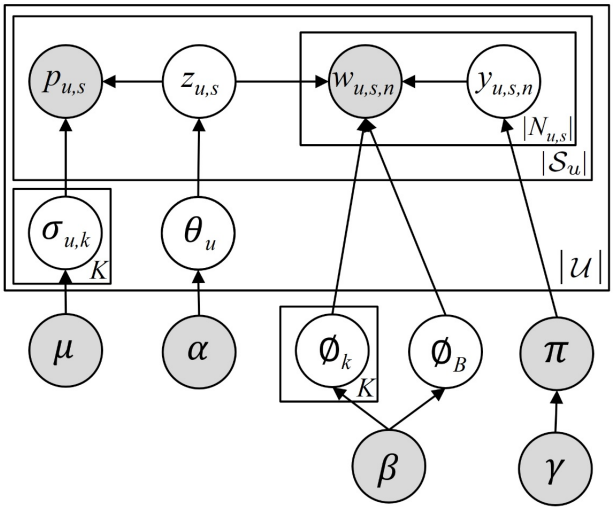


Figure 5.3: Plate diagram of MultiLDA model

Based on the above assumption, we design MultiLDA model with plate diagram shown in Figure 5.3, to simulate the generation of observed users' content from their hidden topical interests and topic-specific platform preference. We assume that there are K topics across all the OSPs. Each topic k has a multinomial distribution ϕ_k over the vocabulary $\mathcal{V}$. We also assume that there is a background topic that captures the background words used across the OSPs. Similarly, this background topic also has a multinomial distribution ϕ_B over the vocabulary. The bias toward the background topic is characterized by a binomial distribution π. To capture users' topical interests, we assume that each user u has a multinomial distribution θ_u over K topics. Lastly, to capture the u's topic-specific platform preference, we assume that, for each topic k, u has a multinomial distribution $\sigma_{u,k}$ over the set of OSPs P. The bias toward the background topic π has Beta prior γ, and the topics' word distributions ϕ_k and ϕ_B have common symmetric Dirichlet prior β. Similarly, users' topic distributions θ_u's and users' topic-specific platform distributions $\sigma_{u,k}$'s have symmetric Dirichlet priors α and μ respectively.

In MultiLDA model, the s-th post of user u is generated as follows. The post's topic $z_{u,s}$ is first chosen by sampling from u's topic distribution θ_u. As

posts are short, we assume that each post has only one topic. The post's content is then generated by sampling its words where each word is sampled independently from the others. For each word $w_{u,s,n}$, a biased coin $y_{u,s,n}$ is flipped to decide where the word is sampled from. The bias of the coin is set to the bias toward the background topic π. The word is sampled from the word distribution of the chosen topic (i.e., $\phi_{z_{u,s}}$) if the coin is *head*, i.e., $y_{u,s,n} = 1$, or that of background topic (i.e., ϕ^B) otherwise. Lastly, the post's platform is chosen by sampling from u's platform distribution specific to the chosen topic, i.e., $\sigma_{u,k}$. The whole generative process of the MultiLDA model is summarized in Algorithm 1.

Algorithm 1 Generative Process for MultiLDA

sample $\phi_B \sim Dir(\beta)$
sample $\pi \sim Beta(\gamma)$
⊡ "Topic Plate"
for topic $k \in \{1, \cdots, K\}$ **do**
    sample the topic's word distribution $\phi_k \sim Dir(\beta)$
end for
⊡ "User Plate"
for user $u \in \mathcal{U}$ **do**
    sample u' topic distribution $\theta_u \sim Dir(\alpha)$
    for topic $k \in \{1, \cdots, K\}$ **do**
        sample u's platform distribution for the topic $\sigma_{u,k} \sim Dir(\mu)$
    end for
    ⊡ "Post Plate"
    for post $s \in \mathcal{S}_u$ **do**
        sample the post's topic $z_{u,s} \sim Multi(\theta_u)$
        ⊡ "Word Plate"
        for word $w_{u,s,n}$ of the post **do**
            sample the word's coin $y_{u,s,n} \sim Bernoulli(\pi)$
            if $y_{u,s,n} = 0$ **then**
                sample the word from background topic $w_{u,s,n} \sim Multi(\phi_B)$
            else
                sample the word from the post's topic $w_{u,s,n} \sim Multi(\phi_{z_{u,s}})$
            end if
        end for
        sample the post's platform $p_{u,s} \sim Multi(\sigma_{u,z_{u,s}})$
    end for
end for

5.3.3 Inference Via Gibbs Sampling

Like in other LDA-based models, the inference problem in the MultiLDA model is intractable [19]. We, therefore, adopt a sampling-based approach to estimate the model's parameters from a given dataset. Specifically, we first randomly initialize the latent topics of posts and latent coins of all words in the dataset. We then use a collapsed Gibbs sampler [76] to iteratively sample the coin for every word and topic for every post. These iterations result in a sample set which allows us to estimate the model's parameters.

Sampling coin for a word. Consider the word $\omega_{u,s,n}$; we denote the bag of coins of all other words by $\mathcal{C}_{-u,s,n}$. Also, we denote the bag of topics of all the posts by $\mathcal{Z}$ and denote the bag of OSPs of all posts by $\mathcal{P}$. The coin $y_{u,s,n}$ is then sampled according to the following equations.

$$p(y_{u,s,n}=0|\mathbf{D}^c_{-u,s,n}) \propto \frac{\mathbf{n_b}(\omega_{u,s,n},\mathbf{D}^c_{-u,s,n})+\beta}{\sum_{\omega\in\mathcal{V}}[\mathbf{n_b}(\omega,\mathbf{D}^c_{-u,s,n})+\beta]}\cdot \frac{\mathbf{n_y}(0,\mathbf{D}^c_{-u,s,n})+\gamma_0}{\mathbf{n_y}(0,\mathbf{D}^c_{-u,s,n})+\mathbf{n_y}(1,\mathbf{D}^c_{-u,s,n})+\gamma_0+\gamma_1} \quad (5.1)$$

$$p(y_{u,s,n}=1|\mathbf{D}^c_{-u,s,n}) \propto \frac{\mathbf{n_w}(\omega_{u,s,n},z_{u,s},\mathbf{D}^c_{-u,s,n})+\beta}{\sum_{\omega\in\mathcal{V}}[\mathbf{n_w}(\omega,z_{u,s},\mathbf{D}^c_{-u,s,n})+\beta]}\cdot \frac{\mathbf{n_y}(1,\mathbf{D}^c_{-u,s,n})+\gamma_1}{(\mathbf{n_y}(0,\mathbf{D}^c_{-u,s,n})+(\mathbf{n_y}(1,\mathbf{D}^c_{-u,s,n})+\gamma_0+\gamma_1)} \quad (5.2)$$

In Equations 5.1 and 5.2, $\mathbf{D}^c_{-u,s,n}$ denotes the tuple ($\mathcal{C}_{-u,s,n}$, $\mathcal{Z},\mathcal{S},\mathcal{P}$, α,β,μ,γ), and $\mathbf{n_y}(c,\mathbf{D}^c_{-u,s,n})$ ($c=0$ or 1) is the number of times in $\mathbf{D}^c_{-u,s,n}$ that words are associated with the coin c. In Equation 5.1, $\mathbf{n_b}(\omega,\mathbf{D}^c_{-u,s,n})$ is the number of times in $\mathbf{D}^c_{-u,s,n}$ that the word ω is associated with the background topic. Similarly, in Equation 5.2, $\mathbf{n_w}(\omega,z,\mathbf{D}^c_{-u,s,n})$ is the number of times in $\mathbf{D}^c_{-u,s,n}$ that the word ω is associated with topic z. In these equations, the first terms on the right hand side are the posterior information of $y_{u,s,n}$,

i.e., the likelihoods that the word $\omega_{u,s,n}$ is generated by the background topic (Equation 5.1) or by topic $z_{u,s}$ (Equation 5.2). The second terms are the prior information of $y_{u,s,n}$, i.e., the likelihood of $y_{u,s,n} = c$ given coins of all other words.

Sampling topic for a post. Now consider the s-th post of user u, we denote the bag of topics of all other posts by $\mathcal{Z}_{-u,s}$. Also, we denote the bag of coins of all the words by $\mathcal{C}$. The topic $z_{u,s}$ is then sampled according to the following equation.

$$p(z_{u,s} = z|\mathbf{D}^z_{-u,s}) \propto \prod_{y_{u,s,n}=1} \frac{\mathbf{n_{wz}}(\omega_{u,s,n}, z, \mathbf{D}^z_{-u,s}) + \beta}{\sum_{w\in\mathcal{V}}[\mathbf{n_{wz}}(\omega, z, \mathbf{D}^z_{-u,s}) + \beta]} \cdot \frac{\mathbf{n_p}(p_{u,s}, z, \mathbf{D}^z_{-u,s}) + \mu}{\sum_{p\in\mathcal{P}}[\mathbf{n_p}(p, z, \mathbf{D}^z_{-u,s}) + \mu]} \cdot \frac{\mathbf{n_z}(z, u, \mathbf{D}^z_{-u,s}) + \alpha}{\sum_{k=1}^{K} \mathbf{n_z}(k, u, \mathbf{D}^z_{-u,s}) + \alpha} \quad (5.3)$$

In Equation 5.3, $\mathbf{D}^z_{-u,s}$ denotes the tuple $(\mathcal{Z}_{-u,s}, \mathcal{C}, \mathcal{S}, \mathcal{P}, \alpha, \beta, \mu, \gamma)$. $\mathbf{n_{wz}}(\omega, z, \mathbf{D}^z_{-u,s,n})$ is the number of times in $\mathbf{D}^z_{-u,s}$ that word ω is associated with topic z. $\mathbf{n_p}(p, z, \mathbf{D}^z_{-u,s})$ is the number of times in $\mathbf{D}^z_{-u,s}$ that posts about topic z are associated with platform p. Lastly, $\mathbf{n_z}(k, u, \mathbf{D}^z_{-u,s})$ is the number of times in $\mathbf{D}^z_{-u,s}$ that posts of user u are associated with topic k. In the equation, the first and second terms on the right hand side are the posterior information of $z_{u,s}$, i.e., the likelihoods that the post's words and platform are generated by the topic z respectively. The third term is the prior information of $z_{u,s}$, i.e., the likelihood of $z_{u,s} = z$ given topics of all other posts.

In our experiments, we used symmetric priors with $\alpha = 50/K$, $\beta = 0.01$, $\mu = 0.01$, and $\gamma_0 = \gamma_1 = 0.01$. Each time, we run the model for 600 iterations of Gibbs sampling. The first 100 iterations were ignored to remove the effect of the random initialization. We take 25 samples with a gap of 20 iterations in the last 500 iterations to estimate the model's parameters.

5.4 Experimental Evaluation

In this section, we perform some experiments to evaluate MultiLDA and to compare with TwitterLDA, the state-of-the-art topic model for short social posts. We first describe the experimental setup and evaluation criteria. Next, the platform choice prediction task is then introduced as part of our evaluation experiments.

5.4.1 Experiment Setup

We evaluate MultiLDA model in two aspects, namely (i) the effectiveness in modeling topics in content from multiple OSPs, and (ii) the accuracy of predicting users' platform choices as they generate posts.

We use the TwitterLDA as our baseline. While TwitterLDA is the state-of-the-art topic model for tweet posts, it can be easily adapted to "tag" posts. It is important to note that TwitterLDA model does not consider platform information associated with the posts. It assumes that all posts are from a single platform.

Training and Test Datasets. For each base user, we randomly selected 80% to 90% of posts of the user to form the training set, and use the remaining posts as the test set. We then learn the MultiLDA and TwitterLDA models using the training set and apply the learned models on the test set.

5.4.2 Post Content Modeling

To evaluate the effectiveness of MultiLDA and TwitterLDA in modeling posts across OSPs, we compute the likelihood of the training set and perplexity of the test set. The model with the higher likelihood or, the lower perplexity is considered superior in the task.

Figure 5.4 shows the likelihood and perplexity achieved by MultiLDA and TwitterLDA as we vary the number of topics K. As expected, as we use a

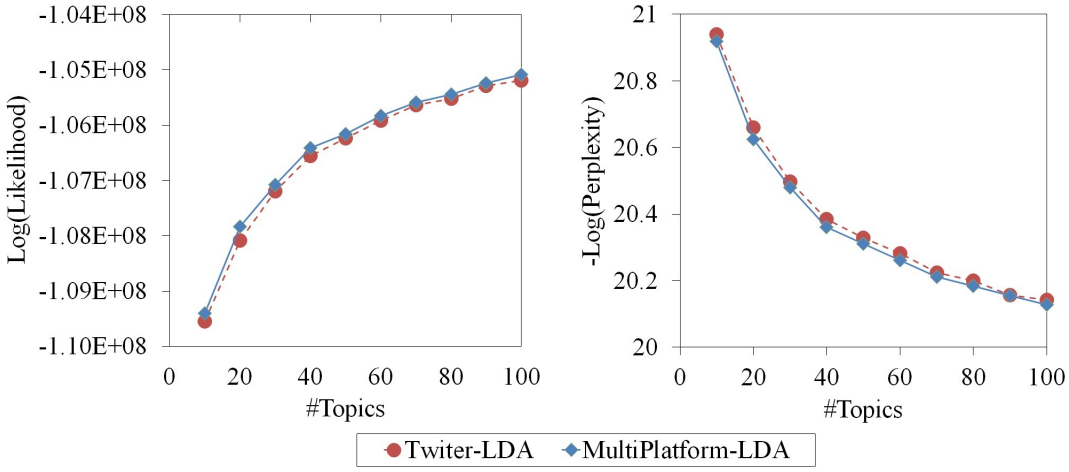


Figure 5.4: Log(Likelihood) and -Log(Perplexity) of MultiLDA and TwitterLDA

larger number of topics, both models achieve a higher likelihood and smaller perplexity. The quantum of improvement, however, reduces as K increases. We notice that the improvement reaches a plateau when K is 80 or above.

The figure also shows that MultiLDA outperforms TwitterLDA in likelihood and perplexity by a very small margin. A possible explanation is our choice of the multiple OSPs datasets which has relatively sufficient data generated by each user. When a user has enough training data from multiple OSPs, TwitterLDA can learn the user topics quite well compared with MultiLDA. It suggests that there are not many users with strong topic-specific platform preferences for MultiLDA to yield much higher likelihood or lower perplexity than TwitterLDA.

5.4.3 Platform Choice Prediction

To evaluate the predictive power of MultiLDA and TwitterLDA, we get them to predict users' platform choices given the content of the test posts. The platform choice of a test post is predicted by MultiLDA by (i) assigning the post's topic using the trained MultiLDA, and then (ii) selecting the most probable platform for the assigned post topic where the user's topic-specific platform distribution determines the most likely platform.

For TwitterLDA which does not model platform choices, we generate the predicted platform choice of a given test post by (i) assigning the particular post's topic using the trained TwitterLDA, and then (ii) returning the most popular platform choice for the assigned topic according to the training set.

We use *Average F1* to measure the accuracy of platform choice prediction results. For each OSP p (i.e., Twitter, Instagram, or Tumblr), we first define its precision, recall and *F1* as follows.

$$Prec_p = \frac{\#\text{ posts with } p \text{ as the correctly predicted platform}}{\#\text{ posts with } p \text{ as the predicted platform}}$$

$$Recall_p = \frac{\#\text{ posts with } p \text{ as the correctly predicted platform}}{\#\text{ posts with } p \text{ as the platform}}$$

$$F1_p = \frac{2 \cdot Prec_p \cdot Recall_p}{Prec_p + Recall_p}$$

We measure $Prec_p$, $Recall_p$ and $F1_p$ by taking average of their values over three runs of prediction each using a different randomly selected training and test sets. By taking the average over three OSPs, we obtain the *Average F1* as $\frac{1}{3}\sum_p F1_p$

Figure 5.5 shows the F1 scores of both MultiLDA and TwitterLDA for each OSP and the average F1 with a number of topics varying from 20 to 100. We also include a baseline which always predicts Twitter (the OSP with most posts) as the platform choice. We observe that MultiLDA outperforms TwitterLDA model in every OSP although the margin is small on the Twitter. On Instagram and Tumblr, MultiLDA significantly performs better than TwitterLDA by more than 50% and 30% respectively. The figure also shows that the prediction results do not change significantly for a different number of topics. Considering all three OSPs, MultiLDA improves the Avg F1 by 30% compared with TwitterLDA.

This good prediction accuracy of MultiLDA suggests that individual-level platform preferences still matter. We will further examine and discuss this in

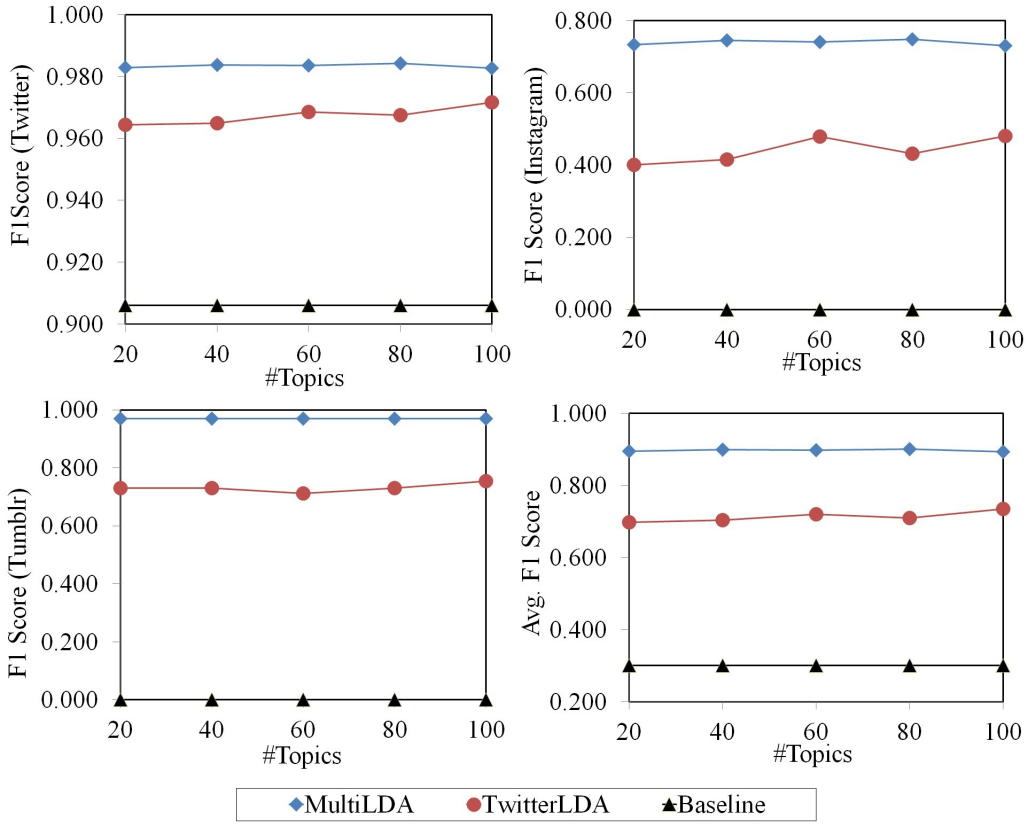


Figure 5.5: F1 scores for Twitter (top left), Instagram (top right), and Tumblr (bottom left), and the average F1 score of the three OSPs (bottom right)

the empirical study section.

5.5 Analysis on Platform Choices and Topics

In this section, we present several empirical findings on user topics and platform choices learned by MultiLDA. First, we analyze the similarity in user's topics across the different OSPs. Next, we compare some of the popular topics shared by the users on the different OSPs. Finally, we examine two case studies that further highlight some of the characteristics of MultiLDA.

5.5.1 Platform Topics Analysis

We analyze the differences (and some similarities) of popular topics among the three OSPs. We will also present two prediction case studies to validate the different approaches of platform choice prediction by MultiLDA and Twit-

terLDA. The number of topics in the MultiLDA model is set to 100 for this empirical analysis.

For any pair of OSPs p_i and p_j, we compute for each user u the Jensen-Shannon Divergence (JSD) between the u's topic distributions on p_i and p_j as follows.

$$JSD(p_i \| p_j | u) = \frac{1}{2} D(p_i \| p_j | u) + \frac{1}{2} D(p_j \| p_i | u)$$

where $D(p_i \| p_j | u)$ is the Kullback-Leibler divergence defined by:

$$D(p_i \| p_j | u) = \sum_k P(k | p_i, u) log \frac{P(k | p_i, u)}{P(k | p_j, u)}$$

where $P(k | p_i, u)$ denotes probability of a topic k when user u posts on platform p_i.

JSD measures how similar a user shares topics at two different OSPs. It returns a value between 0 and 1. A JSD score of 1 means that the user has identical topic distribution on both OSPs. A zero JSD score means completely different topic distributions are shared on the two OSPs.

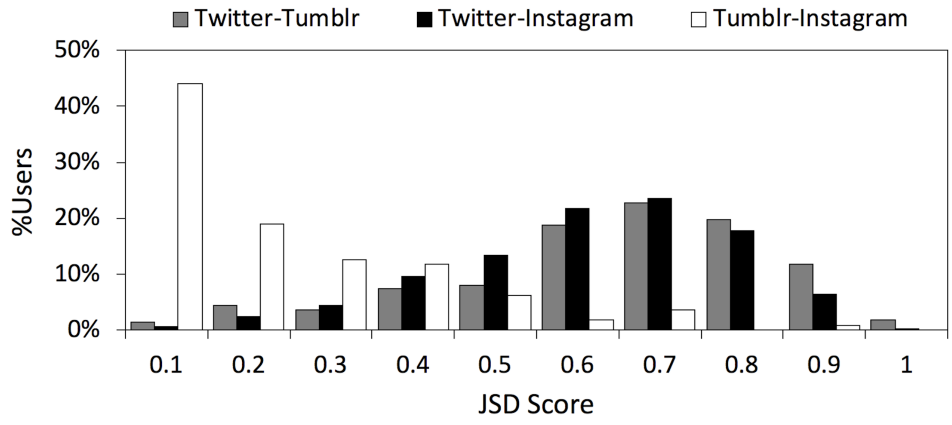


Figure 5.6: JSD score distributions of users for (Twitter, Instagram), (Twitter, Tumblr) and (Instagram,Tumblr)

Figure 5.6 depicts the JSD score distribution of users having accounts on different OSP pairs. The figure shows that most users enjoy higher JSD (or higher topic distribution similarities) between Twitter and Instagram and be-

tween Twitter and Tumblr. Even so, there are very few users with JSD more than 0.8. Among users with Instagram and Tumblr posts, most of them see much smaller topic distribution similarity. In fact, there are many of them having JSD ≤ 0.1.

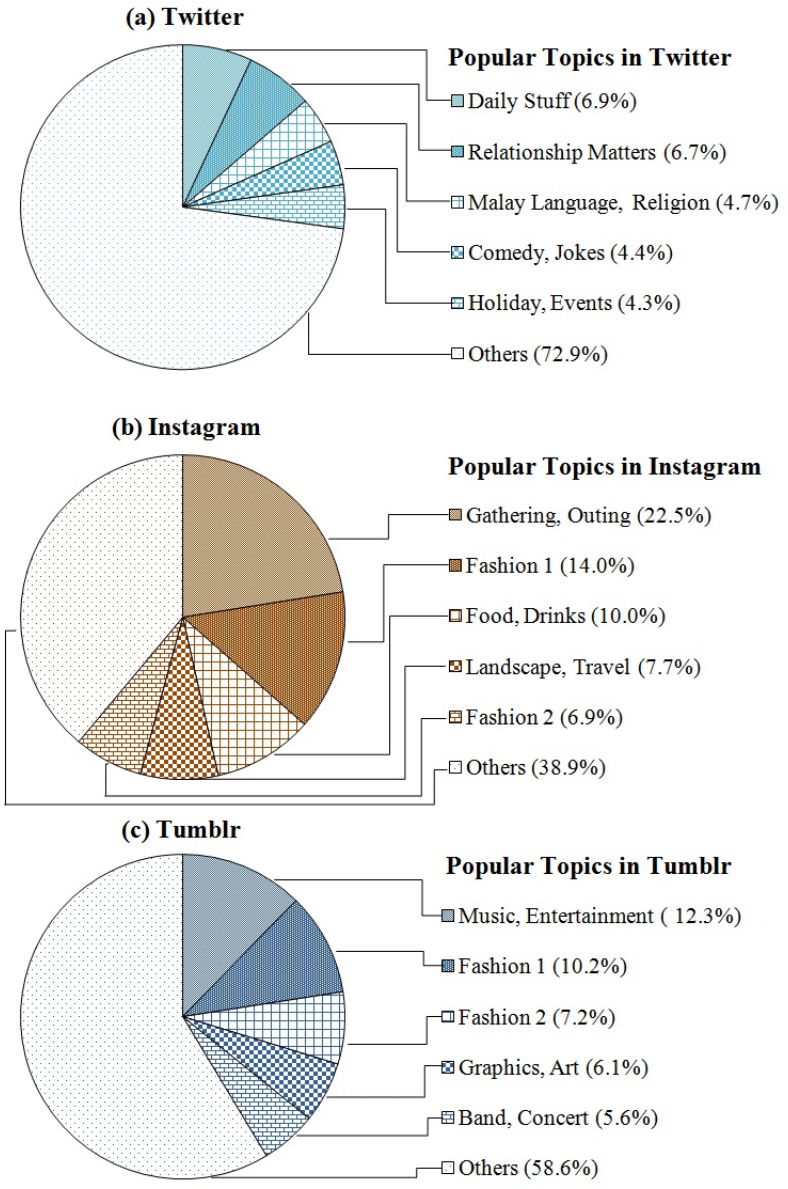


Figure 5.7: The proportion of top topics in (a) Twitter, (b) Instagram, and (c) Tumblr

Figure 5.7 shows the top five popular topics among the base users' posts in (a) Twitter, (b) Instagram and (c) Tumblr. The labels of the topics are manually assigned after examining the topics' top words. When two topics are very similar, we add numbers behind the topic labels (e.g., "*Fashion 1*",

"*Fashion 2*", etc.) to distinguish them. The number in parentheses represents the topic likelihood value. For each topic, the top words are those having the highest likelihoods given the topic, and the top posts are those having the lowest perplexities given the topic.

From the charts, we notice some differences among the popular topics of the three OSPs. In particular, the popular topics on Twitter are very different from those on Instagram and Tumblr. The popular topics in Twitter are about daily chatters while the popular topics on Instagram and Tumblr tend to be more visual (e.g., *Fashion* and *Landscape*). Instagram and Tumblr are observed to share some common popular topics (e.g., Fashion), but there are also some notable differences. For example, topics such as *Music and Entertainment* are popular on Tumblr but not on Instagram. On the other hand, topics such as *Gatherings, Food, and Drinks* are popular on Instagram but not on Tumblr.

The differences in popular topics of the three OSPs suggest that the users could be using each OSP for different purposes (e.g., a user uses Twitter for news sharing but sharing posts about their pop idols in Tumblr). Another explanation could be due to the difference in the networks of friends in different OSPs. In chapter 3, we found that most users prefer to maintain different friendships in different OSPs while keeping only a small clique of common friends across OSPs. Thus, the content shared might cater to the diverse audience from different OSPs.

5.5.2 Case Studies

Case Study 1: Individual User Preferences. As discussed in the earlier section, the presence of individual user's platform preferences enables MultiLDA model to outperform TwitterLDA model. Among the users in our dataset, we found *User1659* who made 95 and 20 posts on Twitter and Instagram respectively. The prediction accuracies for *User1659*'s posts are 0.916 and 0.083 for MultiLDA and TwitterLDA respectively. The accuracy difference

is significantly large. As we examine into the posts of *User1659*, we found that many of the user's Twitter posts fall into the *Music and Entertainment* topic which is popular on Tumblr. Hence, TwitterLDA model wrongly predicted most of *User1659*'s posts to be on Tumblr. However, there are only a few such cases in our dataset. The majority (87%) of the base users in our dataset have their posts predicted with more than 0.7 prediction accuracy using the TwitterLDA model.

Case Study 2: Advantage of Popular Topics in Platforms. Although the MultiLDA model was able to outperform the TwitterLDA model on most users' platform choice prediction, there are a few instances where TwitterLDA outperforms MultiLDA by a small margin. For example, in *User2709*'s platform choice predictions, TwitterLDA achieved a prediction accuracy of 1.0 while MultiLDA achieved a prediction accuracy of 0.875. We examine the two wrong predictions made by MultiLDA and found that the two posts are published on Tumblr, and they fall into the "*Music and Entertainment*" topic. As *User2709* had not published posts on this topic on Tumblr in the training set, MultiLDA was not able to learn and predict the platform choice correctly. Conversely, TwitterLDA had predicted the platform choice correctly as "music and entertainment" is a popular topic on Tumblr. There are very few (< 5 instances) of such exceptions in our dataset. However, this points to exciting future work of extending MultiLDA to use a combination of global and user preferences.

5.6 Summary

In this chapter, we proposed a novel topic model known as MultiPlatform-LDA (MultiLDA), which jointly models OSP topics as well as platform preference of users. We evaluated MultiLDA using real-world datasets from three OSPs and benchmarked against the state-of-the-art topic model. Our experiment results

have shown that MultiLDA outperforms TwitterLDA in both topic modeling and platform choice prediction tasks. We have also empirically shown that users exhibited different topical interests across OSPs and the different OSPs have different popular topics.

Chapter 6

Modeling Topic-Specific Influential Users in Multiple Online Social Platforms

Finding influential users in online social platforms (OSPs) is an important problem with many possible useful applications. HITS and other link analysis methods, in particular, have been often used to identify hub and authority users in web graphs and OSPs. These works, however, have not considered the topical aspect of links in their analysis. Furthermore, most of these works are confined to identifying influential users within a single OSPs. In this chapter [66], we propose two topic-based model: (i) Hub and Authority Topic model (HAT) and (ii) Multiple Platform Hub and Authority Topic model (MPHAT) to identifying topic-specific hub and authority users in single and multiple OSPs respectively. We evaluate HAT and MPHAT against several existing state-of-the-art methods in three tasks: (i) modeling of topics, (ii) platform choice prediction, and (iii) link recommendation.

6.1 Introduction

Online social platforms (OSPs), such as Facebook, Twitter, and Instagram, have grown phenomenally in recent years. It was reported that as of August 2017, Facebook has over 2 billion monthly active users, while Instagram and Twitter have over 700 million and 300 million monthly active user accounts respectively [1]. The vast amount of content and social data generated by these platforms has made them important resources for marketing campaigns such as the diffusion of advertising messages and promotion of new products. Identifying influential users in OSPs is therefore critical to these marketing applications.

Many research works have proposed methods to identify influential users in OSPs. For example, some works determine users' social influence by network centrality measures [25, 12, 56, 61]. Other works adapted HITS [58] and PageRank [95] algorithms which have initially been proposed to determine hub and authority web pages through analyzing the link structure of a web graph to identify influential users in OSPs [103, 114, 127]. Nevertheless, these existing works are either not topic specific or confined to identifying influential users within a single OSP.

Topic and platform specificities are important when analyzing the hub and authority users as they provide more insights about users and reveal in which OSP they are influential. To illustrate the usefulness of topic specificity, consider an example of two users, u_1 and u_2, sharing similar ego network structures. HITS will assign u_1 and u_2 similar authority and hub scores. However, if u_1 is a popular food content contributor who is followed by many food-loving users, while u_2 is a prominent politician followed by many users interested in politics, it is more appropriate to infer that u_1 and u_2 are authority users on food-related and political topics respectively. Platform specificity is also important in identifying influential users across multiple OSPs. Suppose a user u_3 posts much food content and is followed by many food-loving users in an

OSP p_1 but is less active in another OSP p_2, i.e., u_3 contributes less content and forms fewer relationships in p_2. While u_3 is regarded as an authority user on food-related topics, her authority on this topic is found in OSP p_1 but not p_2.

The benefits of studying topic and platform-specific hub and authority users are manifold. Firstly, it enables better user recommendation. For example, when a jazz-loving user joins an OSP, we can recommend her to follow authority users in jazz music in that OSP. Secondly, identifying topic and platform-specific hub and authority users enhances the effectiveness of marketing campaigns. For example, a food and beverage company can reach out to food-related topics authority users across multiple OSPs to promote their products. These are users who can more effectively disseminate food-related marketing messages and influence others in food choices. The opinions of these influential users on competing restaurants and food products are also important feedback to the company. Also, the company can find new food-related authorities in different OSPs referenced by the platform-specific food-topical hub users.

Our main contributions in this work consist of the following.

- We propose two topic-based models, Hub and Authority Topic model (HAT) and Multiple Platforms Hub and Authority Topic model (MPHAT). To the best of our knowledge, HAT is the first model that jointly learns user topics, hub and authority in an OSP, while MPHAT is the first model that learns topic-specific hub and authority users across multiple OSPs.

- We apply the HAT and MPHAT models on real-world datasets and demonstrate that (a) HAT and MPHAT are comparable to state-of-the-art topic models in learning topics from user-generated content, and (b) HAT and MPHAT outperform other models in user link recommendation tasks for both single and multiple platform settings. Empirically, we also

applied HAT and MPHAT to identify topic-specific hubs and authorities within and across Instagram and Twitter.

- We also conduct experiments on synthetic datasets to verify the effectiveness of MPHAT in identifying platform-specific topical hubs and authorities under different dataset parameter settings.

The rest of this chapter is organized as follows: Section 6.2 describes the generative process of our two proposed models. Section 6.3 and 6.4 present the experimental evaluations that we have conducted on real-world and synthetic datasets respectively. The empirical study on the real-world data using HAT and MPHAT models will also be discussed in Section 6.3. Finally, we conclude the chapter in Section 6.5.

6.2 Proposed Models

In this section, we describe our two proposed model: (i) Hub and Authority Topic models (HAT) and (ii) Multiple Platform Hub and Authority Topic model (MPHAT) in detail. We begin by introducing the key elements of the models and their notation. Next, we present the principles behind designing the models and their generative processes. We then present an algorithm for learning the models' parameters and a data sub-sampling strategy to reduce the computational cost.

6.2.1 Notations and Preliminaries

We summarize the main notations in Table 6.1. We use $\mathcal{U}$ to denote the set of users, U and V to denote the sets of followers and followees of all users in $\mathcal{U}$ respectively. For each user $u \in \mathcal{U}$, we denote her posts by S_u. Here, we adopt the bag-of-words representation for each post: that is, each post is represented as a multi-set of words, and the word order is not important. The number of words of the s-th post of user u is then denoted by $N_{u,s}$, while the n-th word

of the s-th post is denoted by $w_{u,s,n}$. Lastly, we denote the word vocabulary by W.

Table 6.1: Notations

Symbol	**Description**
$\mathcal{U}/U/\ V$	Sets of users, followers, and followees
$\mathcal{W}$	Vocabulary of words in users' content, and $\|\mathcal{W}\| = W$
S_u	Sets of posts by user u
$N_{u,s}$	Sets of words in post s_u
$w_{u,s,n}$	n-th word of the s-th post by user u
K	Number of topics
τ_k	Word distribution of topic k
X_u	Topic vector of user u
$\eta_{u,k}$	Platform preference vector of user u for topic k
$p_{u,s}$	Platform of s-th post of user u
H_u	Topic-specific hub vector of user u
A_v	Topic-specific authority vector of user v
$r_{u,v,p}$	Relationship between u and v in platform p $= 1$ if u follows v in platform p, $= 0$ otherwise
γ	Dirichlet priors of τ_k
α, β, σ, δ	Prior shape of $X_{u,k}$, $\eta_{u,k,p}$, A_v, and H_u respectively
κ, ϕ	Prior scale of $X_{u,k}$ and $\eta_{u,k,p}$ respectively

In this work, we adopt a topic modeling approach for modeling users' interests, platform preferences, hubs and authorities specific to each topic. Our proposed models, HAT and MPHAT, consist of the following model elements.

Topic. A topic is a semantically coherent theme of words found in the user posts. Formally, a topic is represented by a multinomial distribution over W (unique) words. For example, a topic about traveling would have high probabilities for words such as *trip*, and *flight*, but low probabilities for other words. Another topic about food would have high probabilities for words such as *coffee* and *sandwich* but low properties for other non-food related words.

Topical interest. This refers to a user's interests for a specific topic. Formally we assign to every user u a topical interest vector $X_u = (X_{u,1}, \cdots, X_{u,K})$ where K is the number of topics and $X_{u,k} \in (0, +\infty)$ for $k = 1, \cdots, K$.

Topic-specific authority. This refers to the authority of a user for a topic. A topic-specific authority user is one who attracts connections from others for the topic she is well known for. We thus assign to every user $v \in V$

a topic-specific authority vector $A_v = (A_{v,1}, \cdots, A_{v,K})$ where K is the number of topics and $A_{v,k} \in (0, +\infty)$ for $k = 1, \cdots, K$.

Topic-specific hub. This refers to users with connections to many other users for specific topics. We assign to every user $u \in U$, a topic-specific hub vector $H_u = (H_{u,1}, \cdots, H_{u,K})$ where K is again the number of topics and $H_{u,k} \in (0, +\infty)$ for $k = 1, \cdots, K$.

Platform preference. For a specific topic k, a user may prefer to share content or connect to other users for topic k in a specific platform that she participates in. We model this user's topical platform preference by assigning to every user u a topic-specific platform preference vector, $\eta_{u,k} = (\eta_{u,k,1}, \cdots, \eta_{u,k,P})$, where P is the number of platforms. Note that the users' platform preferences are only modeled in MPHAT.

6.2.2 Model Design Principles

The HAT model is designed to generate user posts and following links based on their topical interests, hubs, and authorities in single OSP environment. MPHAT extends HAT by also considering the users' platform preferences when generating user posts and following links in multiple OSPs environment. We employ topic modeling approach similar to LDA [19] and Twitter-LDA [141] for generating user posts from topics. We also use a factorization approach to generate the following links from topic-specific platform preferences, hubs, and authorities.

The notable point in our models is in the explicit and direct modeling of the relationships among topical interests, platform preferences, hubs, and authorities. In HAT and MPHAT, user topical interests and platform preferences not only determine post content and which platform the content will be shared but also play essential roles in determining hubs and authorities. The relationships are however not deterministic, but probabilistic in nature. The HAT and MPHAT models recognize that it is necessary for a user to be interested in a

topic before she becomes an authority or hub for that topic. However, a user who has a keen interest in a topic may not be authority or hub for that topic. Moreover, different topical hub or authority users can be found on different platforms. HAT and MPHAT models, therefore, learn for each user the numerical scores of her topic-specific hub and authority. Also, unlike the existing models that return scores normalized across users, topics, or platforms, HAT and MPHAT aim at learning users' explicit, unnormalized scores, which can be used directly or normalized when required.

6.2.3 Generative Process

We depict the plate diagram of the HAT and MPHAT models in Figure 6.1 and 6.2 respectively. The generative processes for HAT and MPHAT are summarized in Algorithm 2 and 3 respectively. Recall that the number of topics K is given, we denote the word distribution of topic k by τ_k and assume that it is sampled from a given Dirichlet prior with parameter γ. HAT and MPHAT then generate the user posts and following links as follows.

Generating topic interest vectors. For both HAT and MPHAT, we first generate the users' topical interest vectors. For each user u (also the user v in the plate diagram), the k-th dimension of her topical interest vector, $X_{u,k}$, is sampled from the Gamma distribution with shape α and scale κ. Gamma distribution is chosen over Gaussian because we want the values of topical interests to be positive values.

Generating topic-specific platform preference vectors. Specific for MPHAT, we follow a similar approach to generate user's topic-specific platform preference vector. Firstly, we assume that there are P OSPs. For every user u and every topic k, the p dimension of u's platform preference vector specific to topic k $\eta_{u,k,p}$ is sampled from the Gamma distribution with shape β and scale ϕ.

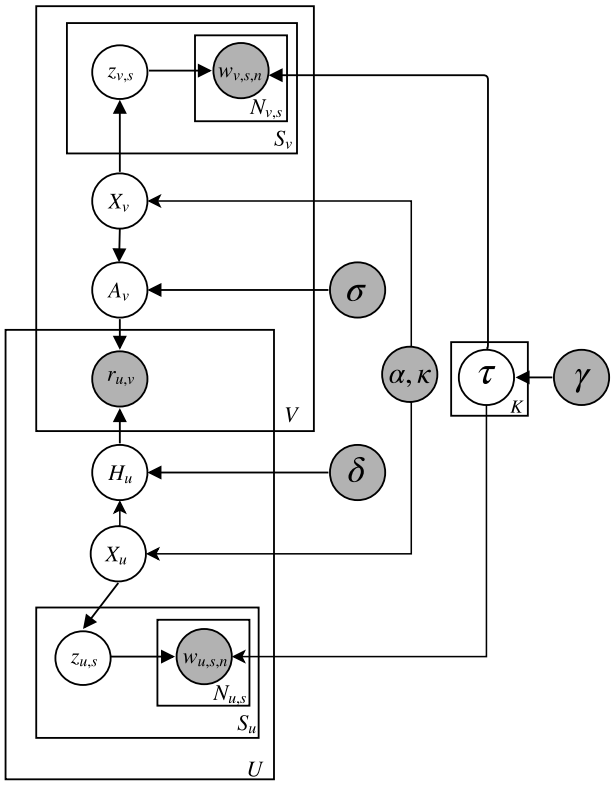


Figure 6.1: Plate Diagram of HAT Model

Generating posts. For HAT and MPHAT to generate the s-th post of user u, the post's topic $z_{u,s}$ is first sampled from the multinomial distribution with parameter $\theta_u = \mathsf{s}(X_u)$. Here $\mathsf{s}(X)$ is the Softmax function[1] that converts an arbitrary vector to a probabilistic vector of the same dimension size. Similar to other previous works on modeling user content in social networks [141], we assume that each post has only one topic as it contains a limited amount of text. The post's content is then generated by sampling its words. Each word $w_{u,s,n}$ is sampled from the word distribution of the chosen topic, i.e., $\tau_{z_{u,s}}$, independently from the other words. For MPHAT, we also sampled the OSP on which the post is shared from the multinomial distribution $\Omega_{u,z_{u,s}} = \mathsf{s}(\eta_{u,z_u})$.

Generating topic-specific hub and authority vectors. HAT and MPHAT incorporate two main ideas in generating user topic-specific hubs and authorities vectors. Firstly, HAT and MPHAT model the users' topic-

[1]https://en.wikipedia.org/wiki/Softmax_function

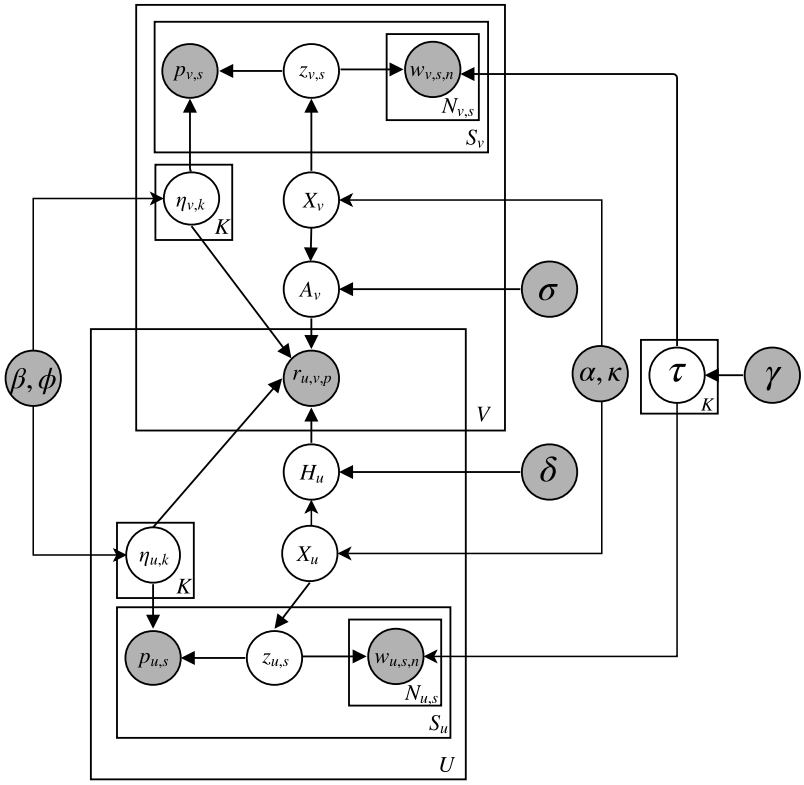


Figure 6.2: Plate Diagram of MPHAT Model

specific hub and authority values as positive numeric values. Secondly, HAT and MPHAT probabilistically relate these hub and authority values to user topical interests. Hence, we propose to model a user's topic-specific hub and authority scores using Gamma distributions whose means are the user's interest in the topics, and the scores will be positive real numbers. Specifically, the topic-specific authority score of user $v \in V$ for topic k, $A_{v,k}$, is sampled from the Gamma distribution with shape σ and scale $\frac{X_{v,k}}{\sigma}$. Similarly, the topic-specific hub score of user $u \in U$ for topic k, $H_{u,k}$, is sampled from the Gamma distribution with shape δ and scale $\frac{X_{u,k}}{\delta}$. Due to the property of Gamma distributions[2], both $A_{v,k}$ and $H_{u,k}$ share the same expectation $X_{u,k}$.

Generating links. In HAT, we generate users' following links in a single OSP. We use $r_{u,v}$ to denote the relationship between u and v: $r_{u,v} = 1$ if u follows v, and $= 0$ otherwise. We sample $r_{u,v}$ from the Bernoulli distribution

[2]https://en.wikipedia.org/wiki/Gamma_distribution

with mean $f(H_u^T A_v, \lambda)$. Here $H_u^T A_v$ is the dot product of H_u^T and A_v and f is the function to scale down it to [0,1) and is defined in Equation 6.1.

In MPHAT, we generate platform-specific users' following in multiple OSPs. We use $r_{u,v,p}$ to denote the relationship between u and v on platform p: $r_{u,v,p} = 1$ if u follows v on p, and $= 0$ otherwise. To generate $r_{u,v,p}$, we first derive the platform-specific authority vector of v on platform p, $A^p_{v,k}$, by taking the element-wise product of A_v and vector $\mathsf{s}(\eta_{v,1,p}, \cdots, \eta_{v,K,p})$. Similarly, the platform-specific hub vector of u on platform p, $H^p_{u,k}$, is defined by the element-wise product of H_u and vector $\mathsf{s}(\eta_{u,1,p}, \cdots, \eta_{u,K,p})$. Finally, we sample $r_{u,v,p}$ from the Bernoulli distribution with mean $f(H_u^{pT} A_v^p, \lambda)$. Here $H_u^{pT} A_v^p$ is the dot product of H_u^{pT} and A_v^p and f is the function to scale down it to [0,1) and is defined in Equation 6.1.

$$f(x, \lambda) = 2(\frac{1}{e^{-\lambda x} + 1} - \frac{1}{2}) \tag{6.1}$$

where $\lambda \in (0, 1)$ is an engineering parameter.

In HAT, the likelihood of forming a following link from u to v is therefore factorized into u's topic-specific hub scores, v's topic-specific authority scores. The likelihood is high when these topic-specific hubs and authorities correlate (i.e., u has high hub in topics that v has high authority), and is low otherwise. Similarly, in MPHAT, the likelihood of forming a following link from u to v is factorized into u's topic-specific hub scores, v's topic-specific authority scores, and their platform preferences. The likelihood is high when these topic-specific hubs, authorities and platform preferences correlate (i.e., u has high hub in topics that v has high authority, and both of them have high preference for the same platform), and is low otherwise.

6.2.4 Model Learning

Given the prior γ, and the parameters α, β, δ, σ, ϕ, κ, and λ, we learn the other parameters in HAT and MPHAT model using maximum likelihood approach.

In other words, to learn HAT and MPHAT models, we solve the optimization problem in Equation 6.2 and 6.3 respectively.

$$\{X^*, A^*, H^*, Z^*, \tau^*\} = \text{arg.max}_{X,A,H,Z,\tau} L(\mathcal{D}|\Psi) \tag{6.2}$$

$$\{X^*, \eta^*, A^*, H^*, Z^*, \tau^*\} = \text{arg.max}_{X,\eta,A,H,Z,\tau} L(\mathcal{D}|\Psi) \tag{6.3}$$

In Equation 6.2, $\Psi = \{X, A, H, Z, \tau, \alpha, \beta, \delta, \sigma, \phi, \kappa, \lambda, \gamma\}$ where X represents for the set of X_u for all users $\{u\}$. A and H are similarly defined. Z represents for the bag of topics of all posts, while τ represents for the set of all topic word distributions $\{\tau_k\}$. Lastly, $L(\mathcal{D}|\Psi)$ is the likelihood function of the observed data $\mathcal{D}$ (i.e., posts and following links) given the value of all the parameters. Equation 6.3 is similarly defined.

Similar to LDA-based models, the problem in Equation 6.2 and 6.3 is intractable [19]. We therefore make use of Gibbs-EM method [18] for learning in HAT and MPHAT models. Specifically, we first randomly initialize X, η, A, H, and τ. We then iteratively perform the following steps until reaching a convergence or exceeding a given number of iterations.

- To sample Z while fixing X, η, A, H, and τ. The topic $z_{u,s}$ is sampled according to the following equation.

$$P(z_{u,s} = k|\theta_u, \tau) \propto \theta_{u,k} \times \prod_{n=1}^{N_{u,s}} \tau_{k,w_{u,s,n}} \tag{6.4}$$

  where, again, $\theta_u = \mathsf{s}(X_u)$

- To optimize X, η, A, H, and τ while keeping Z unchanged. In this step, we make use of the alternating gradient descent method [22]. That is, we iteratively optimize X, η, A, H, or τ while fixing all the others.

6.2.5 Parallelization

As suggested by Equation 6.4, the sampling of a post's topic is independent of that of all the other posts. Hence, we can use multiple child processes, each corresponding to a small set of users, to sample the topics for the users' posts simultaneously. Also, in the alternating steps for optimizing X, we can parallelize the computation as the optimization of a user's topic interest vector is independent of that of all other users' topic interest vectors. Similarly, we can parallelize the alternating optimization of A, H, η, and τ.

In our implementation, in sampling Z, we build a process pool and submit a process for sampling topic for posts of $\frac{1}{N}$ of the users where N is the pool's size. In the ideal case, we can reduce the running time of sampling Z to N times. Similarly, we use the process pool to reduce the running time in the alternating optimization steps.

6.2.6 Data Sub-Sampling

Like previous factorization and mixed membership models, the HAT and MPHAT models consider both link and non-link relationships of all pairs of users. This consideration makes the overall complexity of the HAT and MPHAT models to be $O(N_u^2)$ where N_u is the number of users, which is not practical for large-scale social networks. We, therefore, choose to use a data sub-sampling method to reduce the computational cost. To do that, for each user u, we keep all u's out links (i.e., the links where u follow other users) and $m\%$ of its out non-links (i.e., the no-links where u does not follow some other users). These $m\%$ non-links are selected from the followees of u's followees (i.e., the 2-hops non-existent links). This selection strategy retains only a subset of relationships that carry strong signal of users' hub and authority values while filtering out the remaining data that may contain noise.

6.3 Experiments on real-world dataset

Ideally, we should evaluate HAT and MPHAT by comparing the authority and hub users identified by the model with ground truth authority and hub users. However, it is difficult to find ground truth in real-world datasets. For such datasets, we evaluate HAT and MPHAT against some baseline methods on three tasks: (i) modeling of topics, (ii) users' platform choice prediction, and (iii) link recommendation. We first introduce the real-world datasets which we have collected for our model evaluation. Next, we describe the experiments conducted and report the results. Finally, we present several empirical findings on the topics, hub and authority users learned by the HAT and MPHAT model.

6.3.1 Dataset

Our model evaluation requires multiple datasets that allow us to observe user topical interests and preferences. Furthermore, as we are interested in studying authorities and hubs across online social networks, we require some users to have accounts on multiple OSPs. Public datasets that satisfy the above requirements are not available. Thus, we specially collect two datasets from two popular OSPs that fulfill our requirements, namely Twitter, a short-text microblogging site, and Instagram, a photo-sharing social media site. Both Twitter and Instagram support directed relationships among users, which reflect the preferences of users towards *following* other authority users. Furthermore, the hub and authority users in the two OSPs may differ concerning different topics.

For Twitter data, we collected a set of Singapore-based Twitter users who declared Singapore locations in their user profiles. These users were identified by an iterative snowball sampling process starting from a small seed set of well known Singapore Twitter users followed by traversing the follow links to other Singapore Twitter users until the sampling iteration did not get any more new

Table 6.2: Statistics for Instagram and Twitter Datasets. Numbers in () refer to counts that involve users with accounts on both OSPs and the links among these accounts only.

	Instagram	Twitter
Total users	5,633 (932)	5,401 (932)
Total links	342,719 (22,529)	276,299 (25,379)
Avg Links/user	60 (24)	51 (27)
Max followers	803 (217)	2,048 (421)
Max followings	672 (147)	991 (172)
Min followers	5 (5)	5 (5)
Min following	5 (5)	5 (5)
Total posts	636,593 (121,856)	944,035 (143,317)
Max posts/user	200 (200)	200 (200)
Min posts/user	10 (40)	40 (40)
Avg posts/user	113 (130)	174 (153)

users. From these users, we obtain a subset of users who are active, i.e., have more than 50 directed links, and posted at least 40 tweets between October and December 2016. Subsequently, we retrieve the posts of these *active* Twitter users. A similar approach is used to retrieve the data of active Instagram users who have more than 50 directed links and posted at least ten posts between October and December 2016.

To identify users who have accounts on both Twitter and Instagram among the above active Twitter users, we obtain a subset of users who mention their Instagram accounts in their Twitter bio descriptions. If a mentioned Instagram account is active and does not exist in our subset of active Instagram users, we retrieve the posts and links of that account and add it to our Instagram user set. A similar approach is used to retrieve users who have mentioned their Twitter accounts in their Instagram bio descriptions. Table 6.2 shows the statistics about the collected datasets. In total, we gathered 5,633 Instagram users and 5,401 Twitter users. Among the gathered users, 932 pairs of Twitter and Instagram user accounts are owned by the same users, i.e., these users have active accounts on the two OSPs.

6.3.2 Experiment Setup

We evaluate HAT and MPHAT models in three tasks, namely, (i) topic modeling, (ii) platform choice prediction, and (iii) link recommendation. The first task focuses on comparing the topics learned by HAT and MPHAT with those learned by other baseline models. The second task applies MPHAT to predict users' platform choices as they publish posts. Finally, the last task applies HAT and MPHAT to the prediction of missing links in OSPs. Note that three evaluation tasks will be conducted in the multiple OSP setting. For example, in the first task, we not only model the topics in individual OSPs (i.e., Twitter and Instagram separately) but also topics across both OSPs. In the second task, we predict the platform choices of users who have accounts on multiple OSPs. Finally, in the last task, we train HAT and MPHAT with user relationships from multiple OSPs and predict links to users in individual OSPs.

6.3.2.1 Baselines

For topic modeling, we compare HAT and MPHAT with LDA [19] and TW_LDA [141]. LDA and TW_LDA are two popular topic models for text documents and Twitter content respectively.

For platform choice prediction, we compare MPHAT with TW_LDA and MultiPlatform-LDA (MultiLDA), which we proposed in Chapter 5. MultiLDA learns the user's platform preferences from their posts. Although TW_LDA does not model platform choices, we could infer the posts' platform based on the popular platform choice for the topics learned using TW_LDA.

For link recommendation, we compare HAT and MPHAT against several baselines: HITS, WTFW, and common user interests learned by LDA and TW_LDA. The intuition for interest-based baselines is that user who shares common interests are likely to follow each other due to homophily [86]. WTFW models the topic-specific and social relationships among users, while HITS

returns the authority and hub scores of users based on the relationship network structure.

6.3.2.2 Parameter Setting

In our experiments, the parameter setting of LDA, TW_LDA, and WTFW methods are set to the default values as recommend in their origin. HITS method is parameter free. For HAT and MPHAT methods, we found that the Gibbs-EM algorithm converges around after 200 alternating iterations, each iteration includes 10 gradient descent steps. Topics' prior is set to a symmetric Dirichlet distribution with $\gamma = 0.001$ as widely used in previous works. Both shape α and scale κ of the Gamma prior of users' topical interest X_{uk} are set to 2 for all users u and all topics k. This setting makes X_{uk}'s mean and standard deviation close to 4 and 3 respectively. That means X_{uk} deviates moderately with respect to its mean, hence, $\mathsf{s}(X_{uk})$ is moderately but not extremely skewed toward any topic. This is reasonable as we expect that it is very less likely that users totally focus on a single topic. Similarly, both shape β and scale ϕ of the Gamma prior of users' platform preference η_{ukp} are set to 2 as we do not expect users, who have an account on multiple OSPs, to totally focus on some single OSP. Also, the shapes σ and δ of Gamma priors of users' authority and hub are set to 2. This makes the means of users' authority A_{uk} and hub $H_{u,k}$ close to their topical interest $X_{u,k}$. The scaling parameter λ is set to 0.01 through empirical evaluation on list values.

6.3.2.3 Evaluation Metrics

Topic modeling evaluation. For evaluation on topic modeling, we compute the likelihood of the training set and perplexity of the test set when HAT, MPHAT, and the baselines are applied to the OSP datasets. The model with higher likelihood and lower perplexity is considered superior in this task.

Platform choice prediction evaluation. For evaluation on platform choice

prediction, we get the models to predict users' platform choices given the content of the test posts. MPHAT predicts the platform choice of a test post by first assigning the posts topic using the trained MPHAT, and then selecting the most probable platform for the assigned post topic where the most probable platform is determined by the users topic-specific platform preference distribution.

For TW_LDA which does not model platform choices, we generate the predicted platform choice of a given test post by first assigning the particular posts topic using the trained TW_LDA, and then returning the most popular platform choice for the assigned topic according to the training set.

Finally, we compute the accuracy of platform choice prediction. *Accuracy* for platform choice prediction is defined as:

$$Accuracy = \frac{\#\text{posts with platform correctly predicted}}{\#\text{posts in all platforms}}$$

Link recommendation evaluation. For evaluation on link recommendation, we first define the link recommendation task as recommending new links to a user in a given OSP, i.e., we want to recommend users, other users, to follow in a specific OSP. Thus, given a user u, we first rank her predicted *following* and *non-following* of a specific OSP in the test set by some link scores. Then, we recommend u other users v who are in the specific platform and are higher on the link scores.

For MPHAT, the link score, $score_{MPHAT(u,v,p)}$ that user u would follow user v is measured by the likelihood that $r_{u,v,p} = 1$ as computed based on the two users' hub, authority, and platform preference as described in Section 6.2.3 on *Generating links*. Similarly, for HAT, the score, $score_{HAT(u,v)}$, is the likelihood that u follows v as computed based on the two users' hub and authority learnt by HAT as described in Section 6.2.3 on *Generating links*.

For HITS, the score is measured by taking the product of u's hub (h_u) and

v's authority (a_v):

$$score_{HITS(u,v)} = h_u \cdot a_v \tag{6.5}$$

For LDA, the score is measured by taking the inner product of the topical interests θ_u and θ_v:

$$score_{LDA} = \sum_{k=1}^{K} \theta_{u,k} \cdot \theta_{v,k} \tag{6.6}$$

The same way is also applied to measure links' scores in TW_LDA. Lastly, for WTFW, we directly use the link scores returned by the model.

Finally, we use *precision at top k* and Mean Reciprocal Rank (MRR) [120] to measure the accuracy of link recommendation. *Precision at top k* is defined as:

$$Prec_k = \frac{\sum_{u \in u_k} |L_u \cap L'_{u,k}|}{k \cdot |u_k|}$$

where u_k is a set of users with at least k positive links, L_u and $L'_{u,k}$ are the set of u's positive links and set of top k predicted links for u.

6.3.2.4 Training and Test Datasets

We generate three pairs of training and test datasets which will be used in our experiments: (i) *Instagram*, (ii) *Twitter* and (iii) *combined* datasets.

For *Instagram* datasets, we randomly select 80% of Instagram posts and links from each user who have an account on Instagram to form the training set and use the remaining posts and links as the test set. A similar process is applied to generate the *Twitter* training and test dataset. The *Instagram* and *Twitter* datasets are used to conduct single platform link recommendation experiments.

For the *combined* datasets, we randomly select 80% of platform-specific posts and links from each user to form the training set and use the remaining posts and links as the test set. When combining the two OSPs, the users who have accounts on both Twitter and Instagram will be unified into a single user

identity. The *combined* datasets are used to conduct multiple OSP setting experiments in the three evaluation tasks, i.e., topic modeling, platform choice prediction, and link recommendation.

6.3.3 Evaluation on Topic Modeling

We evaluate the topic modeling of HAT, MPHAT and the baselines on three datasets mentioned in Section 6.3.2.4. Figure 6.3 shows the likelihood and perplexity achieved by HAT, MPHAT, LDA, and TW_LDA. As expected, the larger the number of topics, the higher the likelihood, and lower perplexity are archived by all models. The quantum of improvement, however, reduces as the number of topics increases.

Figure 6.3 also shows that HAT and MPHAT outperform LDA, and are comparable to TW_LDA in the topic modeling task. This result supports the insights from previous work which suggested that standard LDA does not work well for short social media text as both Instagram photo captions and Twitter tweets are much shorter than normal documents [141]. A possible explanation for the similar results achieved by HAT, MPHAT, and TW_LDA can be due to the three models assuming that each post has only one topic.

Interestingly, we also observe that HAT, MPHAT and TW_LDA have outperformed LDA more in Twitter than Instagram. A possible explanation can again be attributed to the different length of the post in different OSPs; Twitter tweets are shorter with a 140 character limit, while Instagram photo captions are longer with no limitation in length imposed.

6.3.4 Evaluation on Platform Choice Prediction

We next evaluate MPHAT and the baselines in a platform choice prediction task using the *combined* dataset. The task predicts the platforms to be used for posts from users with accounts on both Instagram and Twitter. Figure 6.4 shows the *accuracy* of MPHAT, MultiLDA, and TW_LDA for each OSP with

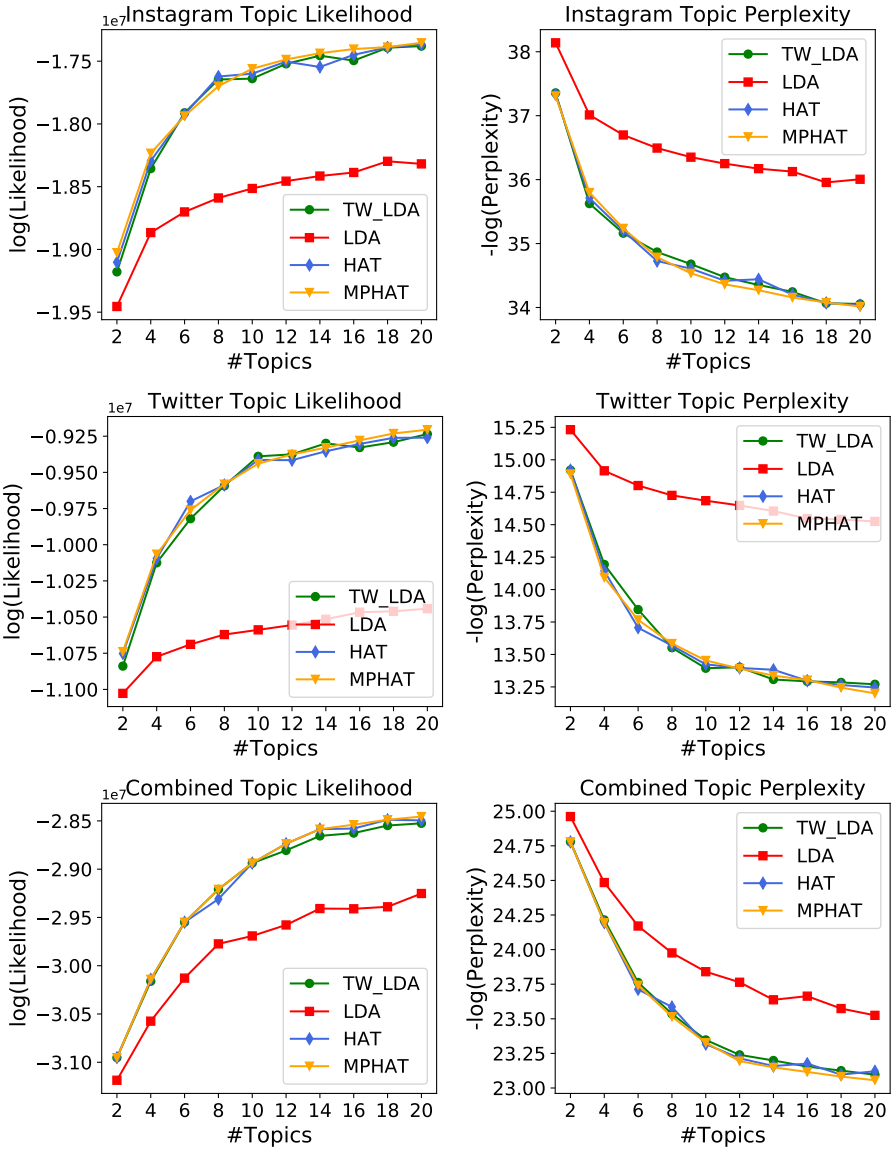


Figure 6.3: Likelihood and perplexity of topics modeled in Instagram, Twitter and combined datasets

the number of topics varying from 12 to 20. We observe that MPHAT and MultiLDA outperform TW_LDA by about 35% in this prediction task. The figure also shows that the prediction results do not change significantly for the different number of topics.

We also observe that MultiLDA outperforms MPHAT by a minimal margin. A possible reason for this observation could be due to the noise introduced by the user relationships; MultiLDA learns the users' platform preference from their posts, while MPHAT considers both users' posts and relationships when learning the users' platform preference. Some users, albeit few, might form

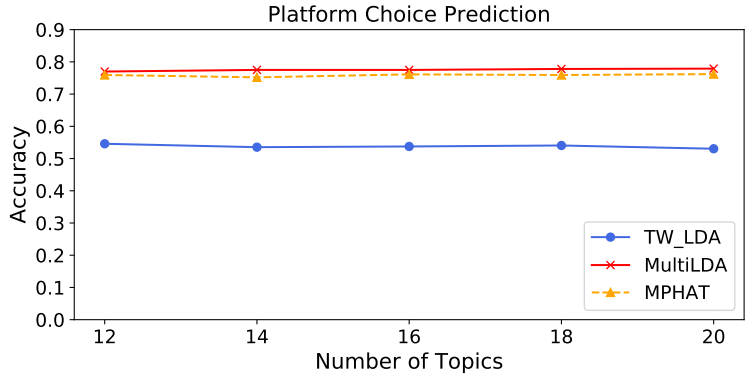


Figure 6.4: Accuracy of platform choice prediction at various number of topics

a lot of relationships in Twitter but they seldom tweet, and this could lead MPHAT to infer that the user has a stronger preference in Twitter.

6.3.5 Evaluation on Link Recommendation

In link recommendation experiments, we consider all links in test datasets as positive instances, and in principle, all the non-existent links as negative instances. Nevertheless, due to the sparsity of OSPs, the number of possible non-links is enormous. Thus, we limit the negative instances to all the nodes which are 2-hops away from the source node of each positive link, which is about 100 times the number of positive instances. The evaluation on link recommendation is conducted in two settings: (i) *multiple platforms* and (ii) *single platform* link recommendation.

In the multiple platforms link recommendation setting, we train HAT, MPHAT and the baseline models on the *combined* training dataset and perform link recommendation in individual OSPs separately using the *combined* test dataset. This experiment aims to evaluate the models when recommending links in multiple OSPs. To further analyze the model effectiveness, we will present the recommendation results involving (a) all types of links and (b) links among users who have accounts on both OSPs (i.e., MP Links) using the *combined* test dataset.

In single platform link recommendation setting, the models are trained

Table 6.3: Multiple platform Instagram and Twitter link recommendations

Method	P@1	P@2	P@3	P@4	P@5	MRR
Instagram						
LDA	0.017	0.017	0.018	0.019	0.020	0.065
TW_LDA	0.015	0.017	0.017	0.017	0.018	0.059
HITS	0.069	0.065	0.057	0.051	0.050	0.135
WTFW	0.086	0.070	0.058	0.052	0.048	0.141
HAT	0.087	0.078	0.073	0.067	0.064	0.160
MPHAT	**0.114**	**0.104**	**0.097**	**0.090**	**0.086**	**0.200**
Twitter						
LDA	0.020	0.019	0.019	0.018	0.017	0.067
TW_LDA	0.017	0.017	0.018	0.019	0.019	0.067
HITS	0.100	0.094	0.084	0.078	0.076	0.203
WTFW	0.152	0.125	0.109	0.100	0.093	0.261
HAT	0.196	0.163	0.144	0.129	0.117	0.305
MPHAT	**0.226**	**0.182**	**0.156**	**0.141**	**0.130**	**0.337**

on a single OSP training dataset, say *Instagram* training dataset, and the link recommendation is performed on the same single OSP test dataset, i.e., *Instagram* test dataset. The purpose of this experiment setting is to evaluate HAT and MPHAT ability in single platform link recommendation compared with other single platform methods.

6.3.5.1 Multiple Platforms Link Recommendation

Table 6.3 shows the multiple platforms link recommendation results for Instagram and Twitter. Note that for HAT, MPHAT and the topic-specific baselines, i.e., WTFW, LDA and TW_LDA, the number of topics learned is set to 18 as beyond which, the quantum of improvement on topic likelihood and perplexity are significantly reduced (see Section 6.3.3).

We observe that MPHAT outperforms HAT and all baselines in both *precision at top k* and MRR for both Instagram and Twitter. When measured by MRR, MPHAT significantly outperforms HITS by more than 50% and 60% on Instagram and Twitter respectively. This result suggests that the topical context is essential in link recommendation. MPHAT also improves the MRR of the common user interests baselines by more than two-fold. This observation also indicates the importance of network information in link recommendation.

Table 6.4: Stratified Instagram and Twitter link recommendations

Method	P@1	P@2	P@3	P@4	P@5	MRR
Instagram						
HAT (All)	0.087	0.078	0.073	0.067	0.064	0.160
MPHAT (All))	0.114	0.104	0.097	0.090	0.086	0.200
%Improvement	31%	31%	32%	33%	33%	25%
HAT (MP)	0.032	0.035	0.037	0.038	0.040	0.096
MPHAT (MP)	0.047	0.065	0.066	0.066	0.063	0.152
%Improvement	43%	84%	77%	72%	57%	59%
Twitter						
HAT (All)	0.196	0.163	0.144	0.129	0.117	0.305
MPHAT (All))	0.226	0.182	0.156	0.141	0.130	0.337
%Improvement	15%	11%	8%	9%	11%	10%
HAT (MP)	0.050	0.057	0.056	0.055	0.052	0.126
MPHAT (MP)	0.073	0.075	0.070	0.062	0.059	0.161
%Improvement	46%	29%	24%	12%	12%	28%

Considering both OSPs, MPHAT and HAT also outperform WTFW by more than 10% in MRR. Interestingly, this demonstrates the importance of hub when modeling topical links; WTFW models susceptibility as users who are interested in a particular topic, while MPHAT and HAT model hub as users who are not only interested in a topic but follow users who are also authority users in that topic. Finally, when measured by MRR, MPHAT outperforms HAT by more than 25% and 10% on Instagram and Twitter respectively. This result demonstrates MPHAT's superiority over HAT in recommending links in multiple OSP setting.

Table 6.4 shows the results for links among users who have accounts on both platforms. We observe that MPHAT has significant improvement over HAT for both *all links* and *MP links*. In particular, MPHAT observes 25% improvement by MRR over HAT for *all links* recommendation in both Instagram and Twitter. This observation could be attributed to MPHAT model design, which considers the users' platform preferences. For example, when user u, who has accounts on both Instagram and Twitter, is an authority for a specific topic k, HAT will recommend other Instagram and Twitter users who are hubs for topic k to follow u. However, suppose u is more active on Instagram. She is more likely to be an authority for topic k on Instagram only.

MPHAT, which models u's platform preferences, would instead recommend only Instagram users who are hubs for topic k to follow u.

We also note that the MRRs for Instagram and Twitter *MP links* are lower than *all links* recommendation for both models. We examined the learned model parameters and found that most of the users who have accounts on both OSPs are topical authorities but not strong hubs. On average, 48.91% of the top 100 authority users across the 18 topics are users on both OSPs. Conversely, only 19.91% of the top 100 hub users across the 18 topics are users on both OSPs. These characteristics of the users on both OSPs make it harder to recommend *MP links* to these users because most of them they are authorities and have less propensity to follow other authorities.

6.3.5.2 Single Platform Link Recommendation

Table 6.5 shows the single platform link recommendation results for Instagram and Twitter. Note that for topic-specific models, the number of topics learned in the training phase is set to 8 and 10 for Instagram and Twitter respectively.

Table 6.5: Single platform Instagram and Twitter link recommendations

Method	P@1	P@2	P@3	P@4	P@5	MRR
Instagram						
LDA	0.018	0.019	0.019	0.019	0.019	0.062
TW_LDA	0.020	0.018	0.017	0.017	0.017	0.059
HITS	0.078	0.070	0.063	0.057	0.054	0.145
WTFW	0.099	0.082	0.071	0.064	0.059	0.167
HAT	0.103	0.092	0.086	0.081	0.078	0.182
MPHAT	**0.123**	**0.113**	**0.106**	**0.100**	**0.097**	**0.211**
Twitter						
LDA	0.017	0.017	0.018	0.019	0.019	0.067
TW_LDA	0.024	0.025	0.025	0.024	0.023	0.080
HITS	0.055	0.066	0.064	0.064	0.065	0.169
WTFW	0.169	0.146	0.132	0.123	0.115	0.296
HAT	0.220	0.166	0.144	0.130	0.120	0.319
MPHAT	**0.220**	**0.182**	**0.159**	**0.146**	**0.135**	**0.335**

Similar to link recommendation in multiple platform setting, we observe that MPHAT outperforms all baselines measured by *both precision at top k*

and MRR for both Instagram and Twitter. This result shows that MPHAT can also perform well in single platform link recommendation.

Interestingly, we also observe that the MRR of single platform link recommendation is higher for most models than that of multiple platform link recommendation. A possible explanation could be the additional noise introduced when we combined the Instagram and Twitter datasets to form the *combined* dataset. For example, when recommending Instagram links in the test dataset, we train the models using the Twitter and Instagram links in the *combined* training dataset. The additional Twitter links might be noise in modeling influence of Instagram users, thus making the Instagram link recommendation task more difficult for multiple platforms. The effect of this additional cross-platform noise is further discussed in an empirical analysis in Section 6.3.6.2.

6.3.6 Empirical Analysis

In this section, we first examine the topic-specific platform preferences of users learned by the MPHAT model. Next, we empirically compare the authority and hub users learned by HITS, HAT, and MPHAT. Note that the analysis is conducted on the *combined* dataset.

6.3.6.1 Topic-Specific Platform Preferences

Other than the users' topical interests, authorities and hubs, MPHAT also learns the topical platform preferences of users on multiple OSPs. Here, we showcase the platform preference of users on Instagram and Twitter. Figure 6.5 shows the distributions of platform preferences of users with accounts on multiple OSPs for four selected sample topics, namely, "sports", "current affairs", "beauty" and "gourmet".

Generally, we observe that the distribution of platform preferences differs across the four topics. This observation supports previous research work

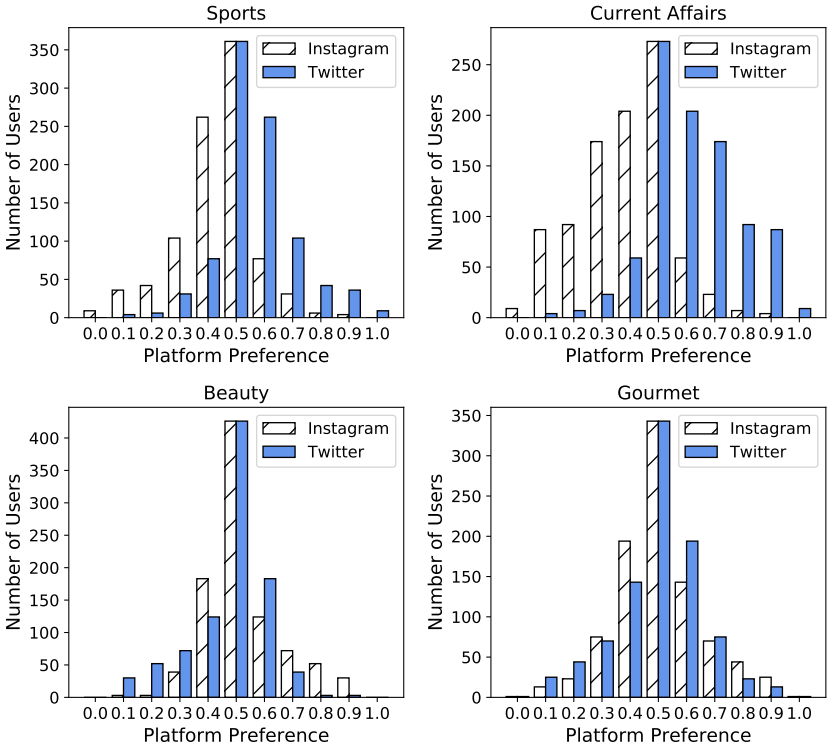


Figure 6.5: Distributions of platform preferences for *sports*, *current affairs*, *beauty*, *gourmet* topics

described in Chapter 5 that suggests that users have different platform preferences for different topics. For example, for the topics on "sports" and "current affairs", the right-leaning bar charts of users' platform preference for Twitter suggest that the users on multiple OSPs prefer to generate their "sports" and "current affairs" content in Twitter, and also link to other Twitter users who have displayed interests on the two topics.

The study on users' topical platform preference also has implications for users' topical authority and hub values. Suppose that "sports" is a popular topic on Twitter and a user, u, who has accounts on both Twitter and Instagram, is identified as a "sports" authority, it is likely that u also has a stronger platform preference for Twitter on "sports" topic. This is because most of the sports-loving users and hubs who follow u are likely to be from Twitter. Note that u may also have other Instagram followers. However, these Instagram followers may not contribute much in determining the u's authority in "sports" topic because majority of the "sports" topical hubs that link to

u are in Twitter. Another empirical example on topical platform preference's effects on topical authority and hub is discussed in Section 6.3.6.2.

6.3.6.2 Hub and Authority Users

Table 6.6 shows samples of the authority and hub users learned by HITS, HAT and MPHAT. HITS determines the authority and hub users strictly by the network structures. Thus, the top authority and hub users identified by HITS are popular Twitter and Instagram users with many followers. On the other hand, MPHAT and HAT can identify authority and hub users for specific topics. For example, for the "sports" topic, MPHAT was able to identify popular football clubs and news media and a sports blogger as top authority users. These users often post sports-related content and are followed by many users interested in sports. Similarly, the top sports topic hub users identified by MPHAT are also sports bloggers and fan group who have followed the sports topic authority users. Similar observations are made in HAT.

Interestingly, we also observe the topic-specific authority and hub users identified by MPHAT different from those that are identified by HAT. Particularly for the topic on "beauty", MPHAT have identified popular lifestyle bloggers who have accounts on both Instagram and Twitter as authority users, while HAT identified cosmetics brands and lifestyle bloggers who only have an Instagram account as authorities. A possible reason for the difference could be the additional cross-platform noise in modeling influence of users with accounts on multiple OSPs, which we have briefly discussed in Section 6.3.5.2.

To investigate this further, we first examine the top 100 hub users for the topic on "beauty" and found that they all have accounts on Instagram. This suggests that "beauty" is a popular topic in Instagram and the authority users followed by these hub users should also have an account in Instagram. Many of these top 100 hub users follow the top 5 "beauty" authority users identified by HAT and MPHAT.

Table 6.6: A sample of authority and hub users in *combined* dataset learned by HITS, HAT and MPHAT. ^{I}@, ^{T}@ and ^{C}@ denotes Instagram, Twitter and multiple OSPs users respectively.

Topic	Top 10 Keywords	Top 5 Authority Users	Top 5 Hub Users
HITS			
-	-	^{C}@xiaxue, ^{T}@blxcknicotine, ^{C}@naomineo_ (lifestyle blogger), ^{C}@benjaminkheng, ^{C}@toshrock (celebrity)	^{T}@blxcknicotine, ^{C}@naomineo_ (lifestyle blogger), ^{C}@benjaminkheng, ^{C}@flyirene (celebrity), ^{T}@herbertrsim (businessman)
HAT			
Beauty	beauty, makeup, skincare, treatment, clozette, collection, lip, foundation, facial, lipstick	^{I}@sephorasg, ^{I}@laneigesg (cosmetics brand), ^{I}@benefitcosmeticssg (lifestyle blogger), ^{I}@beautifulbuns_sg (fashion magazine), ^{I}@thewowoshop (cosmetics ecommerce)	^{I}@sephorasg, ^{I}@etudehousesingapore, ^{I}@laneigesg (cosmetics brand), ^{I}@benefitcosmeticssg (lifestyle blogger), ^{I}@a_must_shop (cosmetic ecommerce)
Sports	game, team, united, arsenal, manutd, league, fans, football, goal, footy_jokes	^{T}@lfc, ^{T}@arsenal (football club), ^{T}@ufc (sports news media), ^{T}@futballtweets, ^{T}@empireofthekop (sports blogger)	^{T}@redsports, ^{T}@empireofthekop, ^{T}@futballtweets, ^{T}@coutinhoflair, ^{T}@theredcardtv (sport blogger)
Current Affairs	business, marketing, digital, trump, tech, ai, data, china, fintech, startup	^{C}@stcom, ^{T}@channelnewsasia (news media), ^{T}@mrbrown (satire blogger), ^{T}@eskimon (businessman), ^{T}@govsingapore (government)	^{T}@wtfsg (satire blogger), ^{T}@eskimon, ^{T}@herbertrsim, ^{T}@alansoon (business), ^{T}@robinhicks_ (editor)
MPHAT			
Beauty	beauty, makeup, skincare, treatment, natural, facial, oil, lip, foundation, clozette	^{C}@jamietyj, ^{C}@bongqiuqiu, ^{C}@bellywellyjelly, ^{C}@Xiaxue, ^{C}@xchubbykitty (lifestyle blogger)	^{I}@ilrpsg (skin-care brand), ^{C}@william82sg, ^{C}@JoannaLHS, ^{I}@makeupforeversg, ^{I}@benefitcosmeticssg (lifestyle blogger)
Sports	arsenal, game, manutd, team, league, football, united, goal, mufc, liverpool	^{C}@stcom, ^{T}@channelnewsasia (news media), ^{T}@lfc , ^{T}@arsenal (football club), ^{T}@redsports (sport blogger)	^{T}@alb_s_fc (football club), ^{T}@redsports, ^{T}@footbalifact, ^{T}@futballtweets (sport blogger), ^{T}@theutdreview (fan group)
Current Affairs	business, marketing, digital, trump, tech, ai, data, china, fintech, startup	^{C}@stcom (news media), ^{T}@eskimon (businessman), ^{T}@mrbrown (satire blogger), ^{C}@papsingapore (political party), ^{T}@govsingapore (government)	^{C}@pinkdotsg (social group), ^{T}@alansoon, ^{C}@skinnylatte, ^{T}@mrscotteddy, ^{C}@mediumshawn (businessman)

However, HAT has given lower authority scores to the users who have accounts on multiple OSPs because other non-hub users in Twitter also follow them, i.e., noise from the links in other OSPs is introduced in HAT's modeling of the users' topical authority. MPHAT mitigates these noise by considering the topical platform preferences of users on multiple OSPs when learning their topical authority and hub scores from the users' links in multiple OSPs. We examined the topical platform preferences of the top 5 "beauty" topic authority users identified by MPHAT and found that these authority users have an average 0.62 platform preferences score for Instagram, i.e., they have a stronger

preference for the Instagram platform on the "beauty" topic. MPHAT weighs the "beauty" topical authority scores of these users by their platform preferences for Instagram and reduces the effect of the noise among the links from Twitter.

6.3.7 Efficiency of Parallel Implementation

We now examine the efficiency of the parallel implementation of the learning algorithm in MPHAT as presented in Section 6.2.5. Figure 6.6 shows the running time of a full iteration of the algorithm when the number of parallel processes is varied from 1 to 20. The figure clearly shows that, as we expected, the running time drops dramatically when the number of parallel processes starts increasing. This shows the efficacy of our parallel implementation. It is also expected that the running time does not decrease significantly after that due to trade off between the actual computing time and the additional time spent on managing the process pool.

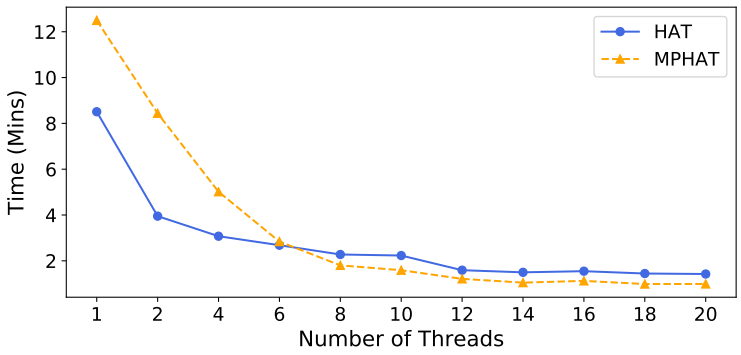


Figure 6.6: Run time of HAT and MPHAT with various number of threads

6.3.8 Data Sub-Sampling Analysis

In Section 6.2.6, we discussed a data sub-sampling method used to reduce the computation cost of HAT and MPHAT. We now empirically examine the effect on link recommendation of the data sub-sampling method. Note that

the experiments are conducted in the *multiple platforms* link recommendation setting described in Section 6.3.5.

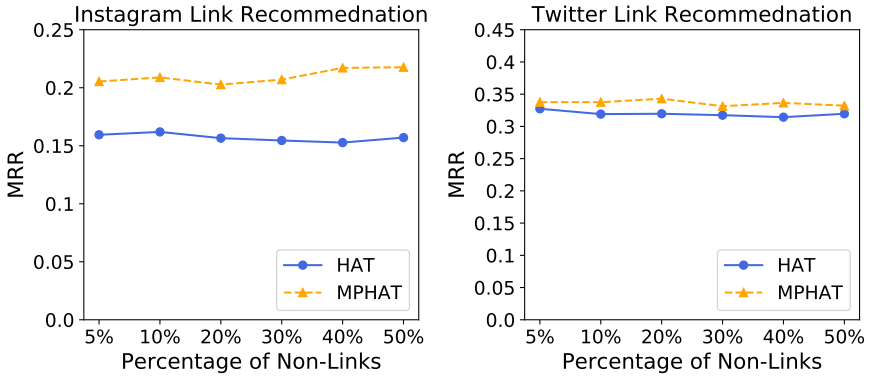


Figure 6.7: MRR for Instagram and Twitter link recommendation with various percentage of non-link sampled

Figure 6.7 shows the HAT and MPHAT's MRR for Instagram and Twitter link recommendation with various percentage of non-link sampled. The link recommendation results are observed to be consistent even when we increase the percentage of non-links sampled for training as the data sub-sampling process is not random but bias to more informative non-links (i.e., followees of users' followees). Thus, the additional less informative non-links would not improve the link recommendation performance significantly.

6.4 Experiments on Synthetic Datasets

In this section, we evaluate the accuracy of HAT and MPHAT using synthetic datasets containing ground truth information. Our goal is also to determine how HAT and MPHAT behave with different data settings. To do this, we need access to the ground truth value of those variables, which is however not available in any real dataset. We, therefore, address this shortcoming by generating synthetic datasets for conducting experimental evaluations.

6.4.1 Synthetic Data Generation

We employ the following steps to generate a dataset with N users on P platforms with posts covering K topics using a vocabulary with W words.

Generating users' topical interest. Given K topics, for each user u, we randomly choose 10% of topics to be ones that u is interested in. That is, the topical interest vector of u, X_u^g, is randomly generated such that the distribution $Softmax(X_u^g)$ (i.e, applying Softmax function on X_u^g) mostly skews on u's interested topics. Also, $\{X_u^g\}_u$ are also normalized across users such that: if users u and v are interested topics k_i and k_j respectively then X_{u,k_i}^g is similarly as large as X_{v,k_j}^g, and they are both much larger than other X_{w,k_l}^g for users w not interested in topics k_l. This normalization does not affect $Softmax(X_u^g)$ but creates clear and distinctive users' topical interest for more accurate comparison among models.

Generating users' platform preference. Given P platforms, as suggested by observations from real datasets used in the Section 6.3, we randomly choose a large subset of users, says 70%, to have accounts on only a single platform, and the remaining users have accounts on all P platforms. For each user u having account on only a single platform, says p, her platform preference vector $\omega_{u,k}^g$ is generated with $Softmax(\omega_{u,k}^g)$ totally focused on the p-th element for any topic k. Otherwise, u has accounts on multiple platforms and $\omega_{u,k}^g$ is defined to have either (i) $Softmax(\omega_{u,k})$ return uniform distribution of platforms u has accounts on, or (ii) $Softmax(\omega_{u,k}^g)$ return a distribution that skews 90% on a certain platform. We generate two synthetic datasets with all the users on multiple platforms either adopting a uniform or skewed platform preference distributions. The two synthetic datasets help to evaluate the models more comprehensively.

Generating users' hub and authority. For each topic k, we randomly choose a small proportion, says q, of users interested in k (refer to the previous step for generating users' topical interest) to be authority users of topic k.

Similarly, q of users interested in k are randomly chosen to be hub users of topic k. As q becomes sufficiently larger, the users who are both authority and hub will increase. If v is among the authority users of k, her authority score $A^g_{v,k}$ is set to $X^g_{v,k}$ plus a small perturbation μ, ($\mu > 0$). Otherwise, $A^g_{v,k}$ is set to be much smaller than $X^g_{v,k}$. Similarly, the hub score $H^g_{u,k}$ of user u on topic k is set in the same way. As users' topical interest $X^g_{*,k}$'s are normalized, A^g_{v,k_i} is similarly as large as A^g_{u,k_j} if v and u are authoritative on topics k_i and k_j respectively. The same observations are held for users' hub scores. These result in a clear separation between authority (or hub) users and non-authority (or non-hub) users in the synthetic datasets. Such a separation helps to evaluate the models more accurately.

Generating topics' word distribution. Given W words in the vocabulary, for each of K topics, its word distribution is randomly generated such that the distribution skews on 10% of the words. Again, this skewness is to create clear and distinctive topics.

Generating the posts and relationships. For each user u, we generate a random number between T_{min} and T_{max} of posts, and for each u's post a random number between L_{min} and L_{max} of words. The posts' topic, words, and following links are generated similar to the generative process described in Section 6.2.3.

6.4.2 Experiment Setup

We evaluate HAT and MPHAT's effectiveness in identifying topical hub and authority users in two synthetic datasets, namely, the *uniform* dataset, which users show no platform preference to generate posts and relationships, and the *skewed* dataset, which the users show platform preferences to generate posts and relationships.

We generate the synthetic datasets with 1000 users ($N = 1000$), 10 topics ($K = 10$) and 2 platforms ($P = 2$). We also set the authority and hub

perturbation μ to 0.1, and the T_{min} and T_{max} for posts generation to be 100 and 200 respectively. For each topic k, $q\%$ of the users who are interested in topic k are also randomly selected as the topical hub and authorities. We vary $q\%$ to be between 10% and 50% in our subsequent experiments. For the learning of the HAT and MPHAT models, we adopt the parameter settings described in S Section 6.3.2.2.

Before evaluating HAT and MPHAT's accuracy in modeling in identifying topical hub and authority users, we first check that the topic learned by the two models are similar to the generated ground truth. To perform this evaluation, for each ground truth topic k_g, we compute the Euclidean distances between k_g's word distribution and the word distributions of the learned topics. The corresponding learned topic which has the smallest Euclidean distance with k_g is assumed to be the matching learned topic k_l.

To evaluate HAT and MPHAT's accuracy in identifying topical hub and authority users, we rank users by the model computed hub and authority scores for each topic, and compare the top $q\%$ users in the ranked lists with the ground truth topical hub and authority users. We measure the model's precision by $Prec_{Auth}@q\% = \frac{\tau_p \cap \tau_g}{\tau_g}$ for each topic k, where τ_p is the set of top $q\%$ authorities predicted by the model and τ_g is the set of authorities in the ground truth. The precision in recovering ground truth topical hubs, $Prec_{Hub}@q\%$, are computed similarly.

6.4.3 Performance Evaluation

We applied HAT and MPHAT on the *uniform* and *skewed* datasets. We first check and match the topics learned by the two models to the ground truth topics by comparing the topics Euclidean distance. Subsequently, we evaluate the accuracy of the two models in topic-specific hubs and authorities ground truth recovery.

Table 6.7: Descriptive stats of HAT and MPHAT matching and learned topics' Euclidean distances.

Method	Min	Max	Mean	StdDev
MPHAT	0.0014	0.0033	0.0021	0.0006
HAT	0.0013	0.0031	0.0018	0.00007

6.4.3.1 Topic Distances Comparison

Table 6.7 shows descriptive statistics of Euclidean distances between the word distributions of topics learned by HAT and MPHAT, and the matching ground truth topics in the *uniform* dataset. We observed that the Euclidean distances between the word distributions of the ground truth topics and topics learned by the two models are small, i.e., the mean distance of 0.0021 and 0.0018 for MPHAT and HAT respectively. This observation suggests that both models can learn the topics well, which is essential for identifying the topic-specific hubs and authorities in the ground truth. Similar observations are made when applied HAT and MPHAT on the *skewed* dataset.

6.4.3.2 Hubs and Authorities Ground Truth Recovery

Figure 6.8 shows the ground truth recovery results for *uniform* and *skewed* datasets. For various q%, we compute the average $Prec_{Auth}@q\%$ and $Prec_{Hub}@q\%$ for 10 topics learned in the two OSPs.

From Figure 6.8, we observed that both MPHAT and HAT performed well in identifying topical hub and authority users in the *uniform* dataset, while MPHAT has outperformed HAT in the *skewed* dataset. The results are reasonable as HAT is designed to identify topical hubs and authorities in a single platform setting, and is thus able to perform well in the *uniform* dataset. It, however, yields poor results for *skewed* dataset. On the other hand, MPHAT performs very well in both data settings. MPHAT learns the users' topical platform preference and thus was able to perform well in identifying the topical hubs an authorities in both synthetic datasets. The results are also consistent across various q%.

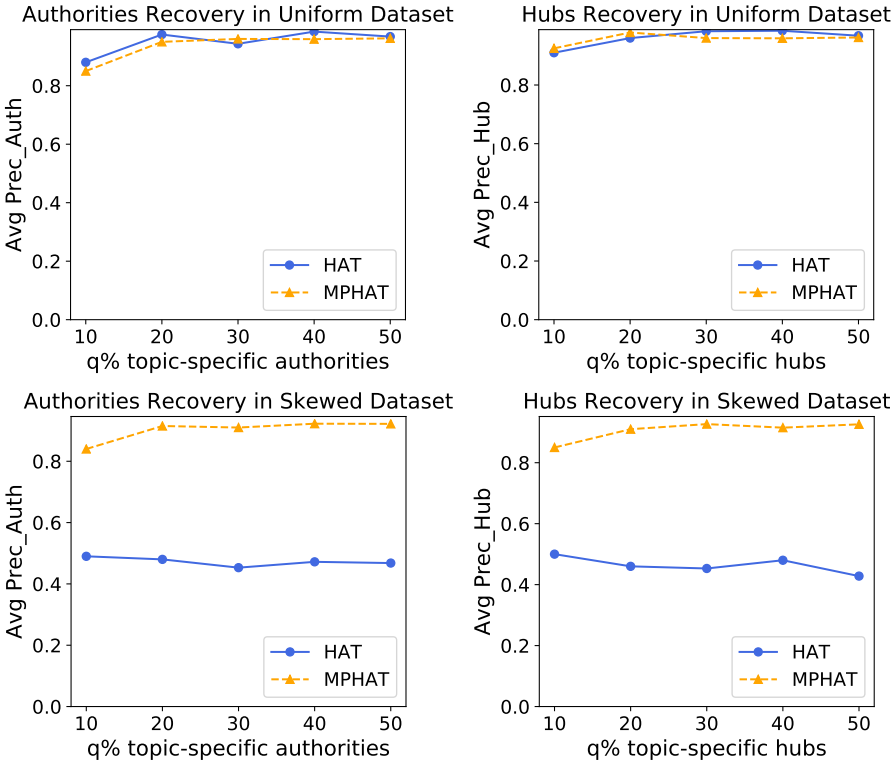


Figure 6.8: $Prec_{Auth}$ and $Prec_{Hub}$ at various $q\%$

6.5 Summary

In this chapter, we have proposed two novel generative models called Hub and Authority Topic Model (HAT) model and Multiple Platform Hub and Authority Topic (MPHAT) model. HAT jointly learns user's topic-specific hubs, authorities, and interests, while MPHAT extends HAT by also learning the users' platform preferences. We evaluated HAT and MPHAT using synthetic and real-world datasets and benchmarked against the state-of-the-art. Our experiments on Twitter and Instagram datasets showed that our proposed HAT and MPHAT outperforms LDA and achieves comparable results as TW_LDA in topic modeling. On platform prediction, MPHAT outperforms the TW_LDA baseline method and can predict which OSP a user would publish his or her posts with reasonable accuracy. On link recommendation, MPHAT outperforms the baseline methods in MRR by at least 10%. We have empirically shown that HAT and MPHAT can identify hub and authority users within and across Twitter and Instagram for different topics. Our experiments on

synthetic datasets also showed that MPHAT outperforms HAT in identifying hub and authority users in multiple OSP setting.

Algorithm 2 Generative Process for HAT Model

⊡ "Generating topics"
for each topic k **do**
    sample the topic's word distribution $\tau_k \sim Dir(\gamma)$
end for
⊡ "Generating user topical interests
for each user u **do**
    for topic $k \in \{1, \cdots, K\}$ **do**
        sample u's interest in topic k: $X_{u,k} \sim \Gamma(\alpha, \kappa)$
    end for
end for
⊡ "Generating user topic-specific authorities and hubs"
for each topic k **do**
    for each user $v \in V$ **do**
        sample v's authority on topic K: $A_{v,k} \sim \Gamma(\sigma, \frac{X_{v,k}}{\sigma})$
    end for
    for each user $u \in U$ **do**
        sample u's hub on topic k: $H_{u,k} \sim \Gamma(\delta, \frac{X_{u,k}}{\delta})$
    end for
end for
⊡ "Generating posts"
for each user u **do**
    for each post s **do**
        sample topic $z_{u,s} \sim Multi(\theta_u)$ where $\theta_u = \mathsf{s}(X_u)$
        for each word slot n **do**
            sample the word $w_{v,s,n} \sim Multi(\tau_{z_{v,s}})$
        end for
    end for
end for
⊡ "Generating following relationship"
for each pair of source user u and target user v **do**
    sample the relationship: $r_{u,v} \sim Bernoulli(f(H_u^T A_v, \lambda))$
end for

Algorithm 3 Generative Process for MPHAT Model

⊡ "Generating topics"
for each topic k **do**
  sample the topic's word distribution $\tau_k \sim Dir(\gamma)$
end for
⊡ "Generating user topical interests and topic-specific platform preferences"
for each user u **do**
  for topic $k \in \{1, \cdots, K\}$ **do**
    sample u's interest in topic k: $X_{u,k} \sim \Gamma(\alpha, \kappa)$
    for platform $p \in \{1, \cdots, P\}$ **do**
      sample u' preference for platform p on topic k: $\eta_{u,k,p} \sim \Gamma(\beta, \phi)$
    end for
  end for
end for
⊡ "Generating user topic-specific authorities and hubs"
for each topic k **do**
  for each user $v \in V$ **do**
    sample v's authority on topic K: $A_{v,k} \sim \Gamma(\sigma, \frac{X_{v,k}}{\sigma})$
  end for
  for each user $u \in U$ **do**
    sample u's hub on topic k: $H_{u,k} \sim \Gamma(\delta, \frac{X_{u,k}}{\delta})$
  end for
end for
⊡ "Generating posts"
for each user u **do**
  for each post s **do**
    sample topic $z_{u,s} \sim Multi(\theta_u)$ where $\theta_u = \mathsf{s}(X_u)$
    for each word slot n **do**
      sample the word $w_{v,s,n} \sim Multi(\tau_{z_{v,s}})$
    end for
    sample platform $p_{u,s} \sim Multi(\Omega_{uz_{u,s}})$ where $\Omega_{uz_{u,s}} = \mathsf{s}(\eta_{u,z_{u,s}})$
  end for
end for
⊡ "Generating following relationship"
for each pair of source user u and target user v **do**
  sample the relationship: $r_{u,v,p} \sim Bernoulli(f(H_u^{pT} A_v^p, \lambda))$
end for

Chapter 7

Conclusion

7.1 Dissertation Summary

With the increased adoption of multiple online social platforms (OSPs) for many types of social and work activities performed by the same population of users, the analysis and modeling of user-generated data from these OSPs is an important yet challenging research task. In this section, we summarize the dissertation work and highlight its main contributions.

Empirical Analysis. To study how users manage their activities across OSPs, we conducted two empirical studies to analyze user-generated data in multiple OSPs. For each study, a multi-OSP dataset is specially gathered such that we can observe the same set of users actively using OSPs over the same period. The novel insights gained from these studies can also be used to derive useful features for prediction tasks that involve multiple OSPs.

In Chapter 3, we proposed a few novel measures to analyze the similarity and evenness of a user's friendship in multiple OSPs. We hypothesize that users have to make their own decisions about whether their online friendships should be fully, partially, or not replicated across different OSPs. The proposed measures, quantifying such user friending preferences, were applied to the empirical analysis of friendships of users who have accounts on Twitter

and Instagram. Our analysis found most of these users maintaining mostly different sets of online friendships across OSPs. Most users appear to be interested in keeping only a very small proportion of common online friends in OSPs. The insights gathered in our user's friendship maintenance empirical study are also used to derive novel user features which can improve friendship link prediction in multiple OSP setting.

In Chapter 4, we proposed another set of novel measures to quantify the similarity in users' topical interests inferred from their collaborative activities within and across multiple OSPs. We applied the proposed measures to the analysis of GitHub and Stack Overflow, which are two popular OSPs used by the software engineering communities. Our analysis found most users displayed similar topical interests in their collaborative activities in the two OSPs, suggesting the possibilities that the users might be using GitHub and Stack Overflow in a complementary manner. We also found that users who perform collaborative activities together tend to share common topical interests in the two OSPs. Using our findings from the empirical study, we proposed a collaborative activity recommendation framework which includes novel user features that allow us to predict a user's collaborative activities in one OSP using the same user's topical interests inferred from his or her collaborative activities in another OSP. Through extensive experiments and case study, we also demonstrated our proposed framework's potential in solving the cold-start problem in collaborative activity recommendations, i.e., recommending activities to a user without knowing the users' past collaborative activity history on the OSP.

User Modeling. In this dissertation, we have focused on modeling the latent user factors related to content and user-user relationships in multiple OSP. The modeling techniques and methodologies proposed in our user modeling tasks enable us to conduct better user profiling and cross-platform empirical studies.

In Chapter 5, we proposed MultiPlatform-LDA (MultiLDA), which extends

LDA for learning users' latent topical interests in multiple OSPs and their topic-specific platform preferences. MultiLDA also allows topics to be shared between different OSPs, thus facilitating the comparison of topical interests across platforms. Through experiments on real-world datasets, we showed that MultiLDA is able to model user topical interests and platform preferences across Twitter, Instagram, and Tumblr. MultiLDA is also generalizable to learn the platform's topics and user's topical interests in single and multiple OSP settings, making it a useful tool for future research that requires learning the latent topics and user's topical interests in OSPs.

In Chapter 6, we proposed two novel generative models, Hub and Authority Topic model (HAT) and Multiple Platform Hub and Authority Topic model (MPHAT), to identity topic-specific influential users in single and multiple OSP settings. Both are extensions of the well-known HITS model which does not consider topics nor multiple OSP setting. Through extensive experiments on real-world and synthetic datasets, we have demonstrated that HAT and MPHAT perform well in (a) topic modeling, (b) platform prediction, and (c) user link recommendation, in both single and multiple OSP settings. Empirically, we have applied HAT and MPHAT to identify topic-specific hubs and authorities within and across Instagram and Twitter. HAT and MPHAT are also generalizable and can be applied to identify influential users and perform social recommendation in different OSPs.

7.2 Future Work

To conclude this dissertation, we discuss some potential future work.

On cross-platform empirical studies, the insights gathered from our empirical studies in Chapters 3 and 4 can be applied to designing and extending modeling techniques to learn latent user factors in multiple OSPs. For example, from our empirical study in Chapter 3 we found that users have preferences

in maintaining different groups of friends in different OSPs while keeping only small cliques of common friends across multiple OSPs. This insight presents the possibility of modeling the latent user's friendship maintenance preferences in multiple OSPs. Currently, the HAT and MPHAT model proposed in Chapter 6 is not able to perform this modeling task because HAT and MPHAT are designed to model user's topic-specific relationship, i.e., user u follows v because u is interested in a topic which v is an authority in, but not the social relationships, i.e., u and v follow each other because they are friends. Thus, we can design new models to learn the latent user's friendship maintenance preferences, which can potentially help to improve social recommendations in OSPs.

Similarly, in Chapter 4 we found that users displayed different topical interests in the various collaborative activities in OSPs, prompting the possibility to model the latent user's activity-specific topical interests within and across OSPs. The MultiLDA model proposed in Chapter 5 is not able to perform this modeling task because MultiLDA is designed to learn the aggregated platform-specific topical interests of a user. Thus, a new model will need to be proposed to address the platform and activity-specific topical interests of users. This new model can help to improve activity recommendation in OSPs.

On user modeling, a natural extension to our work in Chapters 5 and 6 is to consider the dynamic aspect of modeling latent user factors in multiple OSPs. For instance, users may change their topical interests in the different platform over time. For example, a user may change their topical interests over time to follow trending topics in Twitter but remain consistent in his or her topical interests in Instagram. We can extend MultiLDA to learn the evolution of user's topical interests in multiple OSPs. The new dynamic topic model helps better profile user's long-term and short-term topical interests across multiple OSPs.

Similar to users' topical interests, users may change their influence across

topics over time. For example, a popular sports athlete, who is an authorized user on sport-related topics maybe gain authority on food-related topics over time as he or she shares more content on healthy eating and is followed by more users who are interested in food-related topics. A user may also change his or her influence across platforms. For example, when a fashion-related authority Twitter user joins Instagram, he or she may gain popularity rapidly in Instagram and become a new fashion-related authority in Instagram because he or she is quickly followed by other fashion-related hub users who have accounts on the two OSPs. MPHAT can be extended to model these dynamic aspects in the user's platform and topic-specific influence. The new dynamic user influence model can help better identify influence users and provide better social recommendations in OSPs.

Lastly, we can continue to adopt the research framework proposed in Section 1.2 to utilize different combinations of user-generated data to conduct new cross-platform empirical studies and design novel user modeling techniques to learn latent user factors in multiple OSP setting. For example, other than the users' social and work activities covered in Chapter 3 and 4, content sharing and information diffusion across multiple OSPs is also an exciting direction for cross-platform empirical studies. For example, we can examine the content sharing activities of users with accounts on multiple OSPs and identify users who are more likely to cross-share information in multiple OSPs. We can also determine what kind of content is more likely to be transferred and cross-shared between OSPs.

Bibliography

[1] Most famous social network sites worldwide as of august 2017, ranked by number of active users (in millions). Technical report, Statista, 2017.

[2] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[3] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S Yu. Identifying the influential bloggers in a community. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 207–218. ACM, 2008.

[4] Abolfazl Aleahmad, Payam Karisani, Maseud Rahgozar, and Farhad Oroumchian. Olfinder: Finding opinion leaders in online social networks. *Journal of Information Science*, 42(5):659–674, 2016.

[5] Mohammad Y. Allaho and Wang-Chien Lee. Increasing the responsiveness of recommended expert collaborators for online open projects. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 749–758. ACM, 2014.

[6] Zeynep Zengin Alp and Şule Gündüz Öğüdücü. Identifying topical influencers on twitter based on user behavior and network topology. *Knowledge-Based Systems*, 141:211–221, 2018.

[7] Isabel Anger and Christian Kittl. Measuring influence on twitter. In *Proceedings of the International Conference on Knowledge Management and Knowledge Technologies*, page 31. ACM, 2011.

[8] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 2012.

[9] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. Ego networks in twitter: an experimental analysis. pages 3459–3464. IEEE, 2013.

[10] Ali Sajedi Badashian, Afsaneh Esteki, Ameneh Gholipour, Abram Hindle, and Eleni Stroulia. Involvement, contribution and influence in github and stack overflow. In *Proceedings of Annual International Conference on Computer Science and Software Engineering (CSSE)*, pages 19–33. IBM Corp., 2014.

[11] Kartik Bajaj, Karthik Pattabiraman, and Ali Mesbah. Mining questions asked by web developers. In *Proceedings for the Conference on Mining Software Repositories (MSR)*, pages 112–121. ACM, 2014.

[12] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 65–74. ACM, 2011.

[13] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Who to follow and why: link prediction with explanations. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1266–1275. ACM, 2014.

[14] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3):619–654, 2014.

[15] Andrew Begel, Jan Bosch, and Margaret-Anne Storey. Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder. *IEEE Software*, 30(1):52–66, 2013.

[16] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Foundations of multidimensional network analysis. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 485–489. IEEE, 2011.

[17] Bin Bi, Yuanyuan Tian, Yannis Sismanis, Andrey Balmin, and Junghoo Cho. Scalable topic-specific influence analysis on microblogs. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 513–522. ACM, 2014.

[18] Jeff A Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[20] Catherine A Bliss, Morgan R Frank, Christopher M Danforth, and Peter Sheridan Dodds. An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, 5(5):750–764, 2014.

[21] Phillip Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, 2(1):113–120, 1972.

[22] Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht. The alternating descent conditional gradient method for sparse inverse problems. *Journal on Optimization*, 27(2):616–639, 2017.

[23] Casey Casalnuovo, Bogdan Vasilescu, Premkumar Devanbu, and Vladimir Filkov. Developer onboarding in github: The role of prior social links and language experience. In *Proceedings of the ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE)*, pages 817–828. ACM, 2015.

[24] Pew Research Center. Social media site usage 2014. Technical report, Jan 2015.

[25] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceeding of the International AAAI Conference on Web and Social Media (ICWSM)*, 2010.

[26] Yi Chang, Lei Tang, Yoshiyuki Inagaki, and Yan Liu. What is tumblr: A statistical overview and comparison. *SIGKDD Explorations Newsletter*, 16(1):21–29, 2014.

[27] Yang Chen, Chenfan Zhuang, Qiang Cao, and Pan Hui. Understanding cross-site linking in online social networks. In *Proceedings of the SNA-KDD Workshop on Social Network Mining and Analysis*, page 6. ACM, 2014.

[28] Yoon-Sik Cho, Greg Ver Steeg, Emilio Ferrara, and Aram Galstyan. Latent space model for multi-modal social data. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 447–458, 2016.

[29] Morakot Choetkiertikul, Daniel Avery, Hoa Khanh Dam, Truyen Tran, and Aditya Ghose. Who will answer my question on stack overflow? In *Proceedings of the Australasian Software Engineering Conference (ASWEC)*, pages 155–164. IEEE, 2015.

[30] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. Social coding in github: Transparency and collaboration in an open software repository. In *Proceedings of the conference on computer supported cooperative work (CSCW)*, pages 1277–1286. ACM, 2012.

[31] Lucas B. L. de Souza, Eduardo C. Campos, and Marcelo de A. Maia. Ranking crowd knowledge to assist software development. In *Proceedings of the International Conference on Program Comprehension*, pages 72–82. ACM, 2014.

[32] Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V Chawla, Jinghai Rao, and Huanhuan Cao. Link prediction and recommendation across heterogeneous social networks. In *Proceeding of the International Conference on Data Mining (ICDM)*, pages 181–190. IEEE, 2012.

[33] RI Dunbar. The social brain hypothesis. *Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews*, 6(5):178–190, 1998.

[34] Emilio Ferrara, Roberto Interdonato, and Andrea Tagarelli. Online popularity and topical interests through the lens of instagram. In *Proceedings of the conference on Hypertext and social media*, pages 24–34. ACM, 2014.

[35] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[36] Linton C Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3), 1978.

[37] Daniel Gayo-Avello. Nepotistic relationships in twitter and their impact on rank prestige algorithms. *Information Processing & Management*, 49(6):1250–1280, 2013.

[38] Rumi Ghosh and Kristina Lerman. Predicting influential users in online social networks. In *Proceedings of the SNA-KDD Workshop on Social Network Mining and Analysis*, 2010.

[39] Debra S. Goldberg and Frederick P. Roth. Assessing experimentally derived interactions in a small world. *Proceedings of the National Academy of Sciences*, 100(8):4372–4376, 2003.

[40] Georgios Gousios. The ghtorrent dataset and tool suite. In *Proceedings for the Conference on Mining Software Repositories (MSR)*, pages 233–236. IEEE, 2013.

[41] Mohamed Guendouz, Abdelmalek Amine, and Reda Mohamed Hamou. Recommending relevant github repositories: a collaborative-filtering approach. *on Networking and Advanced Systems*, page 34, 2015.

[42] Dongyan Guo, Jingsong Xu, Jian Zhang, Min Xu, Ying Cui, and Xiangjian He. User relationship strength modeling for friend recommendation on instagram. *Neurocomputing*, 239:9–18, 2017.

[43] Weiyu Guo, Shu Wu, Liang Wang, and Tieniu Tan. Social-relational topic model for social networks. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 1731–1734. ACM, 2015.

[44] Tuan-Anh Hoang and Ee-Peng Lim. On joint modeling of topical communities and personal interest in microblogs. In *International Conference on Social Informatics (SocInfo)*, pages 1–16. Springer, 2014.

[45] Tuan-Anh Hoang and Ee-Peng Lim. Microblogging content propagation modeling using topic-specific behavioral factors. *Transactions on Knowledge and Data Engineering (TKDE)*, 28(9):2407–2422, 2016.

[46] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[47] Juan Hu, Yi Fang, and Archana Godavarthy. Topical authority propagation on microblogs. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 1901–1904. ACM, 2013.

[48] Chaoran Huang, Lina Yao, Xianzhi Wang, Boualem Benatallah, and Quan Z Sheng. Expert as a service: Software expert recommendation via knowledge domain embeddings in stack overflow. In *Proceedings of the International Conference on Web Services (ICWS)*, pages 317–324. IEEE, 2017.

[49] Mingqing Huang, Guobing Zou, Bofeng Zhang, Yanglan Gan, Susu Jiang, and Keyuan Jiang. Identifying influential individuals in microblogging networks using graph partitioning. *Expert Systems with Applications*, 102:70–82, 2018.

[50] Lamjed Ben Jabeur, Lynda Tamine, and Mohand Boughanem. Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. In *International Symposium on String Processing and Information Retrieval*, pages 111–117. Springer, 2012.

[51] Jin Yea Jang, Kyungsik Han, Patrick C Shih, and Dongwon Lee. Generation like: Comparative characteristics in instagram. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 4039–4042. ACM, 2015.

[52] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. In *Pro-*

ceedings of the WebKDD and SNA-KDD workshop on Web mining and social network analysis, pages 56–65. ACM, 2007.

[53] Jyun-Yu Jiang, Pu-Jen Cheng, and Wei Wang. Open source repository recommendation in social coding. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1173–1176. ACM, 2017.

[54] Xin Jin and Yaohua Wang. Research on social network structure and public opinions dissemination of micro-blog based on complex network analysis. *Journal of Networks*, 8(7):1543, 2013.

[55] Georgios Katsimpras, Dimitrios Vogiatzis, and Georgios Paliouras. Determining influential users with supervised random walks. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 787–792. ACM, 2015.

[56] Imrul Kayes, Xiaoning Qian, John Skvoretz, and Adriana Iamnitchi. How influential are you: detecting influential bloggers in a blogging community. In *International Conference on Social Informatics (SocInfo)*, pages 29–42. Springer, 2012.

[57] Alexy Khrabrov and George Cybenko. Discovering influence in communication networks using dynamic graph analysis. In *Proceedings of the International Conference on Social Computing (SocialCom)*, pages 288–294. IEEE, 2010.

[58] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5), 1999.

[59] Tamara G Kolda, Brett W Bader, and Joseph P Kenny. Higher-order web link analysis using multilinear algebra. In *Proceeding of the International Conference on Data Mining (ICDM)*. IEEE, 2005.

[60] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. Inferring anchor links across multiple heterogeneous social networks. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 179–188. ACM, 2013.

[61] Nicolas Kourtellis, Tharaka Alahakoon, Ramanuja Simha, Adriana Iamnitchi, and Rahul Tripathi. Identifying high betweenness centrality nodes in large social networks. *Social Network Analysis and Mining*, 3(4), 2013.

[62] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 591–600. AcM, 2010.

[63] Sebastian Labitzke, Irina Taranu, and Hannes Hartenstein. What your friends tell others about you: Low cost linkability of social network profiles. In *Proceedings of the International Workshop on Social Network Mining and Analysis*, pages 1065–1070, 2011.

[64] Ka-Wei Roy Lee and Ee-Peng Lim. Friendship maintenance and prediction in multiple social networks. In *Proceedings of the Conference on Hypertext and Social Media*, pages 83–92. ACM, 2016.

[65] Roy Ka-Wei Lee, Tuan-Anh Hoang, and Ee-Peng Lim. On analyzing user topic-specific platform preferences across multiple social media sites. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 1351–1359, 2017.

[66] Roy Ka-Wei Lee, Tuan-Anh Hoang, and Ee-Peng Lim. Discovering hidden topical hubs and authorities in online social networks. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 378–386. SIAM, 2018.

[67] Roy Ka-Wei Lee and Ee Peng LIM. Measuring user influence, susceptibility and cynicalness in sentiment diffusion. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 411–422. Springer, 2015.

[68] Roy Ka-Wei Lee and David Lo. Github and stack overflow: Analyzing developer interests across multiple social collaborative platforms. In *International Conference on Social Informatics (SocInfo)*, pages 245–256. Springer, 2017.

[69] Roy Ka-Wei Lee and David Lo. Wisdom in sum of parts: Multi-platform activity prediction in social collaborative sites. In *Proceedings of the ACM Conference on Web Science (WebSci)*, pages 77–86, 2018.

[70] William Leibzon. Social network of software development at github. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1374–1376. IEEE, 2016.

[71] Xiang Li, Shaoyin Cheng, Wenlong Chen, and Fan Jiang. Novel user influence measurement based on user interaction in microblog. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 615–619. ACM, 2013.

[72] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[73] Bang Hui Lim, Dongyuan Lu, Tao Chen, and Min-Yen Kan. #mytweet via instagram: Exploring user behaviour across multiple social networks. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 113–120. ACM, 2015.

[74] Antonio Lima, Luca Rossi, and Mirco Musolesi. Coding together at scale: Github as a collaborative social network. *ArXiv*, 2014.

[75] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: An unsupervised approach to link users across communities. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 495–504. ACM, 2013.

[76] Jun S Liu. The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, 1994.

[77] Li Liu, William K. Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1774–1780, 2016.

[78] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 199–208. ACM, 2010.

[79] Xinyue Liu, Hua Shen, Fenglong Ma, and Wenxin Liang. Topical influential user analysis with relationship strength estimation in twitter. In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, pages 1012–1019. IEEE, 2014.

[80] Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. Learning to predict reciprocity and triadic closure in social networks. *Transactions on Knowledge Discovery from Data (TKDD)*, 7(2):5, 2013.

[81] M Magnani and L Rossi. The ml-model for multi-layer social networks. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 5–12. IEEE, 2011.

[82] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. Predict anchor links across social networks via an embedding approach. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1823–1829, 2016.

[83] Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 889–892. ACM, 2013.

[84] Pasquale de Meo, Emilio Ferrara, Fabian Abel, Lora Aroyo, and Geert-Jan Houben. Analyzing user behavior across social sharing environments. *Transactions on Intelligent Systems and Technology (TIST)*, 5(1):14, 2013.

[85] Matthew Michelson and Sofus A Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data*, pages 73–80. ACM, 2010.

[86] Lynn Smith-Lovin Miller McPherson and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[87] Manuela Montangero and Marco Furini. Trank: Ranking twitter users according to specific topics. In *Proceedings of the Consumer Communications and Networking Conference (CCNC)*, pages 767–772. IEEE, 2015.

[88] Marti Motoyama and George Varghese. I seek you: Searching and matching individuals in social networks. In *Proceedings of the International Workshop on Web Information and Data Management*, pages 67–75. ACM, 2009.

[89] Xin Mu, Feida Zhu, Ee-Peng Lim, Jing Xiao, Jianzong Wang, and Zhi-Hua Zhou. User identity linkage by latent user space modelling. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1775–1784. ACM, 2016.

[90] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Symposium on Security and Privacy*, pages 173–187. IEEE, 2009.

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

[92] Yuanping Nie, Yan Jia, Shudong Li, Xiang Zhu, Aiping Li, and Bin Zhou. Identifying users across social networks based on dynamic core interests. *Neurocomputing*, 210:107–115, 2016.

[93] Ermelinda Oro, Clara Pizzuti, Nicola Procopio, and Massimo Ruffolo. Detecting topic authoritative social media users: A multilayer network approach. *Transactions on Multimedia*, 20(5):1195–1208, 2018.

[94] Raphael Ottoni, Diego B Las Casas, Joao Paulo Pesce, Wagner Meira Jr, Christo Wilson, Alan Mislove, and Virgilio Almeida. Of pins and tweets: Investigating how users behave across image-and text-based social networks. In *Proceeding of the International AAAI Conference on Web and Social Media (ICWSM)*, 2014.

[95] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[96] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, pages 45–54. ACM, 2011.

[97] Olga Peled, Michael Fire, Lior Rokach, and Yuval Elovici. Entity matching in online social networks. In *Proceedings of the International Conference on Social Computing (SocialCom)*, pages 339–344. IEEE, 2013.

[98] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How unique and traceable are usernames? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–17. Springer, 2011.

[99] Minghui Qiu, Feida Zhu, and Jing Jiang. It is not just what we say, but how we say them: Lda-based behavior-topic model. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 794–802. SIAM, 2013.

[100] Mohammad Masudur Rahman, Chanchal K Roy, and Jason A Collins. Correct: code reviewer recommendation in github based on cross-project and technology experience. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 222–231. IEEE, 2016.

[101] Baishakhi Ray, Daryl Posnett, Vladimir Filkov, and Premkumar Devanbu. A large scale study of programming languages and code quality in github. In *Proceedings of the ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE)*, pages 155–165. ACM, 2014.

[102] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. Finding expert users in community question answering. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 791–798. ACM, 2012.

[103] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2011.

[104] Daniel Mauricio Romero and Jon Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *Proceeding of the International AAAI Conference on Web and Social Media (ICWSM)*, 2010.

[105] Christoffer Rosen and Emad Shihab. What are mobile developers asking about? a large scale study using stack overflow. *Empirical Software Engineering*, 21(3):1192–1223, 2016.

[106] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[107] Giulio Rossetti, Michele Berlingerio, and Fosca Giannotti. Scalable link prediction on multidimensional networks. In *Proceeding of the International Conference on Data Mining Workshops (ICDMW)*, pages 979–986. IEEE, 2011.

[108] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260. ACM, 2002.

[109] Moshen Shahriari and Mahdi Jalili. Ranking nodes in signed social networks. *Social Network Analysis and Mining*, 4(1):172, 2014.

[110] Yilin Shen and Hongxia Jin. Controllable information sharing for user accounts linkage across multiple online social networks. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 381–390. ACM, 2014.

[111] Kai Shu, Suhang Wang, Jiliang Tang, Reza Zafarani, and Huan Liu. User identity linkage across online social networks: A review. *SIGKDD Explorations Newsletter*, 18(2):5–17, 2017.

[112] Arlei Silva, Sara Guimarães, Wagner Meira Jr, and Mohammed Zaki. Profilerank: finding relevant content and influential users based on information diffusion. In *Proceedings of the Workshop on Social Network Mining and Analysis (SNA-KDD)*, page 2. ACM, 2013.

[113] Georg Simmel. *The Sociology of Georg Simmel*. Simon and Schuster, 1950.

[114] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 807–816. ACM, 2009.

[115] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 659–666, 2004.

[116] Ferdian Thung, Tegawendé F Bissyandé, Daniel Lo, and Lingxiao Jiang. Network structure of social coding in github. In *CSMR*, pages 323–326. IEEE, 2013.

[117] Jason Tsay, Laura Dabbish, and James Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 356–366. ACM, 2014.

[118] Jorge Valverde-Rebaza and Alneu de Andrade Lopes. Structural link prediction using community information on twitter. In *International Conference Computational aspects of social networks (CASoN)*, pages 132–137. IEEE, 2012.

[119] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Proceeding of the International Conference on Social computing (SocialCom)*, pages 188–195. IEEE, 2013.

[120] Ellen M Voorhees and L Buckland. Overview of the trec 2003 question answering track. In *TREC*, volume 2003, pages 54–68, 2003.

[121] Jan Vosecky, Dan Hong, and Vincent Y Shen. User identification across multiple social networks. In *Proceeding of the International Conference on Networked Digital Technologies*, 2009.

[122] Jian Wang, Jiqing Sun, Hongfei Lin, Hualei Dong, and Shaowu Zhang. Convolutional neural networks for expert recommendation in community question answering. *Science China Information Sciences*, 60(11), 2017.

[123] Lin Wang, Bin Wu, Juan Yang, and Shuang Peng. Personalized recommendation for new questions in community question answering. In *Proceeding of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 901–908. IEEE, 2016.

[124] Shaowei Wang, David Lo, and Lingxiao Jiang. An empirical study on developer interactions in stackoverflow. In *Proceedings of the Annual ACM Symposium on Applied Computing*, pages 1019–1024. ACM, 2013.

[125] Wei Wang, Haroon Malik, and Michael W. Godfrey. Recommending posts concerning api issues in developer q&sa sites. In *Proceedings for the Conference on Mining Software Repositories (MSR)*, pages 224–234. IEEE, 2015.

[126] Stanley Wasserman. *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press, 1994.

[127] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the International*

Conference on Web Search and Data Mining (WSDM), pages 261–270. ACM, 2010.

[128] Wei Xie, Cheng Li, Feida Zhu, Ee-Peng Lim, and Xueqing Gong. When a friend in twitter is a friend in life. In *Proceedings of the Web Science Conference*, pages 344–347. ACM, 2012.

[129] Congfu Xu, Xin Wang, and Yunhui Guo. Collaborative expert recommendation for community-based question answering. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-KDD)*, pages 378–393. Springer, 2016.

[130] Jiejun Xu, Ryan Compton, Tsai-Ching Lu, and David Allen. Rolling through tumblr: characterizing behavioral patterns of the microblogging platform. In *Proceedings of the Conference on Web Science (WebSci)*, pages 13–22. ACM, 2014.

[131] Yuto Yamaguchi, Tsubasa Takahashi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Turank: Twitter user ranking based on user-tweet graph analysis. In *International Conference on Web Information Systems Engineering*, pages 240–253. Springer, 2010.

[132] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. Cqarank: Jointly model topics and expertise in community question answering. In *Proceedings of the International on Conference on Information and Knowledge Management (CIKM)*, pages 99–108. ACM, 2013.

[133] Yue Yu, Huaimin Wang, Gang Yin, and Charles X Ling. Reviewer recommender of pull-requests in github. In *Proceedings of the International Conference on Software Maintenance and Evolution (ICSME)*, pages 609–612. IEEE, 2014.

[134] Yue Yu, Huaimin Wang, Gang Yin, and Tao Wang. Reviewer recommendation for pull-requests in github: What can we learn from code review and bug assignment? *Information and Software Technology*, 74:204–218, 2016.

[135] Yue Yu, Gang Yin, Huaimin Wang, and Tao Wang. Exploring the patterns of social behavior in github. In *Proceedings of the International Workshop on Crowd-Based Software Development Methods and Technologies*, pages 31–36. ACM, 2014.

[136] Reza Zafarani and Huan Liu. Connecting users across social media sites: Behavioral-modeling approach. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 41–49. ACM, 2013.

[137] Haochen Zhang, Min-Yen Kan, Yiqun Liu, and Shaoping Ma. Online social network profile linkage. In *Asia Information Retrieval Symposium*, pages 197–208. Springer, 2014.

[138] Lingxiao Zhang, Yanzhen Zou, Bing Xie, and Zixiao Zhu. Recommending relevant projects via user behaviour: an exploratory study on github. In *Proceedings of the International Workshop on Crowd-based Software Development Methods and Technologies*, pages 25–30. ACM, 2014.

[139] Yongzheng Zhang and Marco Pennacchiotti. Predicting purchase behaviors from social media. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 1521–1532. ACM, 2013.

[140] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S. Yu. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1485–1494. ACM, 2015.

[141] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Proceeding of the European Conference on Information Retrieval (ECIR)*, pages 338–349. Springer, 2011.

[142] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, and Xiaoming Li. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *Transactions on Knowledge and Data Engineering (TKDE)*, 28(5):1147–1159, 2016.

[143] Xiaoping Zhou, Xun Liang, Haiyan Zhang, and Yuefeng Ma. Cross-platform identification of anonymous identical users in multiple social media networks. *Transactions on Knowledge and Data Engineering (TKDE)*, 28(2):411–424, 2016.

[144] Jie Zou, Ling Xu, Weikang Guo, Meng Yan, Dan Yang, and Xiaohong Zhang. Which non-functional requirements do developers focus on? an empirical study on stack overflow using topic analysis. In *Proceedings for the Conference on Mining Software Repositories (MSR)*, pages 446–449. IEEE, 2015.