User Information Modelling in Social Communities and Networks

Baoguo Yang

PhD

University of York

Computer Science

September 2015

Abstract

User modelling is the basis for social network analysis, such as community detection, expert finding, etc. The aim of this research is to model user information including user-generated content and social ties.

There have been many algorithms for community detection. However, the existing algorithms consider little about the rich hidden knowledge within communities of social networks. In this research, we propose to simultaneously discover communities and the hidden/latent knowledge within them. We focus on jointly modelling communities, user sentiment topics, and the social links.

We also learn to recommend experts to the askers based on the newly posted questions in online question answering communities. Specifically, we first propose a new probabilistic model to depict users' expertise based on answers and their descriptive ability based on questions. To exploit social information in community question answering (CQA), the link analysis is also considered. We also propose a user expertise model under tags rather than the general topics.

In CQA sites, it is very common that some users share the same user names. Once an ambiguous user name is recommended, it is difficult for the asker to find out the target user directly from the large scale CQA site. We propose a simple but effective method to disambiguate user names by ranking their tag-based relevance to a query question.

We evaluate the proposed models and methods on real world datasets. For community discovery, our models can not only identify communities with different topic-sentiment distributions, but also achieve comparable performance. With respect to the expert recommendation in CQA, the unified modelling of user topics/tags and abilities are capable of improving the recommendation performance. Moreover, as for the user name disambiguation in CQA, the proposed method can help question askers match the ambiguous user names with the right people with high accuracy.

Contents

List of Figures

List of Tables

Acknowledgements

I wish to thank my supervisor Dr. Suresh Manandhar for all the freedom, advice and the kind help he gave to me.

I am also very grateful to my internal examiner Dr. Daniel Kudenko and my external examiner for their suggestions and feedback.

Many thanks go to all the rest people in the artificial intelligence group for their friendship and help. Additionally, I would like to thank all other staffs in our department who provide support to me.

I want to express my acknowledgements to the University of York and China Scholarship Council for their funding support, which help me concentrate on my research.

I also would like to show my big thanks to my neighbours for their help during my study.

Last but not least, special thanks to my great parents and brother for their continuous support and unconditional love in every aspect of my life.

Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of the University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

Some of the material contained in this thesis has appeared in the following published papers. For each published item the primary author is the first listed author.

- Baoguo Yang and Suresh Manandhar. User name disambiguation in community question answering. In Proceedings of Recent Advances in Natural Language Processing (RANLP), pages 707-713, 2015.

- Baoguo Yang and Suresh Manandhar. Exploring user expertise and descriptive ability in community question answering. In Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pages 320-327. IEEE, 2014.

- Baoguo Yang and Suresh Manandhar. Tag-based expert recommendation in community question answering. In Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pages 960-963. IEEE, 2014.

- Baoguo Yang and Suresh Manandhar. STC: A joint sentiment-topic model for community identification. In PAKDD Workshops: Trends and Applications in Knowledge Discovery and Data Mining, pages 535-548. Springer, 2014.

- Baoguo Yang and Suresh Manandhar. Community discovery using social links and author-based sentiment topics. In Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pages 580-587. IEEE, 2014.

Chapter 1

Introduction

This chapter starts by introducing the problems needing to be addressed, which is associated with the research topic and motivations of this thesis. Then it briefly describes the scope of our research work and some hypothesis, followed by the contributions of this thesis. Finally, it provides the overview and outline of this thesis to show the organization of the rest chapters.

1.1 Research Problems and Motivations

The rapid growth of social media has been providing us more and more chance to contact with other people and share our interests and opinions online. There have been increasing number of social media platforms, such as Facebook, Myspace, Twitter, etc. It is very natural that these platforms have been a part of daily life of people worldwide, especially for the younger generations. Accordingly, huge volume of social networking data is being generated everyday, which is not only the user-generated content but also the social link data. Such data is associated with users' behaviors and characters. Hence it is very necessary and interesting to model user information based on the networked data.

1.1.1 The Problems

In recent years, complex networks have been widely studied in many research domains, such as engineering, computer science, biology, etc. Community detection in networks is a popular research direction in the literature. Intuitively, a community can be described as a set of entities, where the entities inside the set are generally closer to each other than to the ones outside the set. For example, in the network of the World Wide Web, communities can be seen as groups of pages associated with related topics.

In social networks, communities are considered as important structures in the form of groups of entities (people). There are many real-world applications for community discovery. For instance, many supermarkets and shopping centers provide online customer communication platforms. The customers can not only post their own reviews about the products and services, but also exchange their opinions and sentiments with others. In order to understand what the customers are caring about and their sentiment, the managers would like to identify groups of users, especially some groups with clear sentiment towards some topics. Note that the customers in each group (community) can discuss multiple topics. According to the major topics and dominant sentiment discussed in some communities, the managers can take measures to address the issues from different communities. Another example is about celebrities' twitters. Generally, the celebrities involve in various social activities and have interpersonal relationships. It is necessary to help them summarize their social circles automatically. More specifically, they would like to get an overview of their social profiles, communities they belong to, topics they discussed, the sentiment towards some topics, and the main people they contacted with in each community. Also, it is very interesting to get the summary and visualization of their related communities during different periods, which enables them to know the evolution of their communities and the people they alienated.

The above two examples mentioned the sentiment information in the networks. Generally, most social networks have sentiment component, which actually plays a non-negligible role for data analysis. For example, the supermarket managers expect to understand the customer satisfaction about the products or services by analysing the networking data. Particularly, those communities with obviously opposite sentiment towards a certain product are worthy of further study.

Twitter, a popular microblogging platform, is not only used by individuals, but also very popular in many organizations, such as companies, hotels, and online supermarkets. As we know many hotels have their own twitter accounts. The customers can send their tweets about opinions and reviews to the hotels, and can comment on other tweets about the environment, food, and service of the hotels. To make full use of the data, it is useful to automatically identify communities associated with this twitter account. The communities with obvious negative polarities should be considered firstly. The hotel managers can take actions to address the main issues these customers proposed, and then response to these groups of people about the quality improvement of the hotel to win more customers, and to avoid the negative information proliferation across communities. Note that if we only extract collections of tweets including same sentiment topics by using traditional sentiment analysis methods instead of mining communities, the important social links will be ignored.

Email is considered as another kind of communication tool, which brings us more convenience to send or receive messages. A huge amount of data are generated online every day. Discovering previously unknown knowledge and relationships among people is very useful and necessary for individuals and organizations. Email is widely used in our daily life, especially in companies and universities. Email correspondence produces abundant social messages associated with social relations. For teachers, their email recipients can be students, colleagues, friends, family members, librarians, and book publishers, etc. To get a high-level overview of the emails in our mailboxes, it is very interesting and necessary to discover our social communities in an automatic way. In each community, we are interested in the topics we discussed, people we contacted with, and the sentiment on some topics. Such information is latent and unobservable.

Identifying communities is very useful and important for various reasons: 1) One can get an initial understanding of the structure of the original networks; 2) Community identification can assist the task of node classification; 3) The detected communities can be used for the decision making.

Community (Social) question answering (CQA) sites, such as Yahoo!Answers[1], Stack Overflow[2], Quora[3] and Baidu Zhidao[4], have attracted increasing user involvement. The rapid growth of these CQA sites brings great convenience for users to communicate. In CQA, one can not only post questions in hope of getting satisfying answers, but also can answer other questions. The popular CQA services like Stack Overflow and Yahoo!Answers have immense user base and Q&A postings. Tens of thousands of new questions involving various topics emerge on the CQA sites every day, some of which are urgent waiting to be answered timely. Such postings can be grouped into different categories (topics), and each category contains some sub-categories (subtopics). Askers and answerers are the main contributors in CQA, where an asker can turn to be an answerer in different Q-A pairs. The expert users are the potential answerers for the arriving questions, which are the pillars, play significant role in CQA. After all, the number of expert users is limited, and their expertise levels on distinct topics are different. Moreover, the expert users are not always online, hence there are still a great number of unanswered questions.

CQA has received increasing attentions in real life, which brings prosperity of the CQA web services. The study of CQA has become a popular research topic, which has brought great

[1] http://answers.yahoo.com/

[2] http://www.stackoverflow.com/

[3] https://www.quora.com/

[4] http://zhidao.baidu.com/

success to CQA services in the past years. In CQA, questions, answers and users are the basic elements to form the question answering network. The expert users are key resources in community question answering sites, which can provide satisfactory answers to the askers and promote the well development of CQA services. In CQA, besides the newly posted questions, there are still a large number of unanswered old questions. Some questions can receive a few answers, whereas few of them are accepted. It is very necessary to enable CQA sites to identify potential experts, recommending similar archived questions or best answers to new questions.

The great success of CQA services comes with the research studies in different directions in community question answering. The major research contains three aspects: expert learning [34, 64, 96, 120], best answer recommendation [37, 45, 121], and similar question retrieval [43, 98, 126].

1.1.2 Motivations

The research on communities has a long history, and it has been paid widely attention in the past decade. In [28, 73], Girvan and Newman propose a popular divisive community detection algorithm based on the concept of betweenness. To improve the speed of the algorithm in [28], a modified algorithm is proposed by Tyler et al. in [101]. Also some overlapping community detection methods has been proposed, like [55, 106]. In addition, dynamic community discovery has been studied in recent years [50, 79], where communities are not static but evolve over time.

However, most of the existing community identification methods intend to learn the community structures just using links, which ignore the content information in social networks. In many social networks, people are not only connected, but also convey messages. Discovering communities by combining link and content has been proposed in the literature [81, 89, 111, 119, 125], however, these methods fail to consider the valuable sentiment information in social networks.

To detect meaningful sentiment-level communities from social networks, an effective discovery model is needed. Whereas existing methods are not suitable for discovering communities with different sentiment-topic distributions.

The main goal of CQA sites is to establish online platform for us to ask questions or answer the questions from others. In StackOverflow, a large number of new questions are posted every day, which require responses from the members of QA communities. However, different people are interested in different areas. Even in the same field, the background knowledge of users can be distinct. In this case, when a new question appears, it is not unusual that the asker needs to wait some time to get the first response to this question until an online user who is familiar with and willing to answer this question.

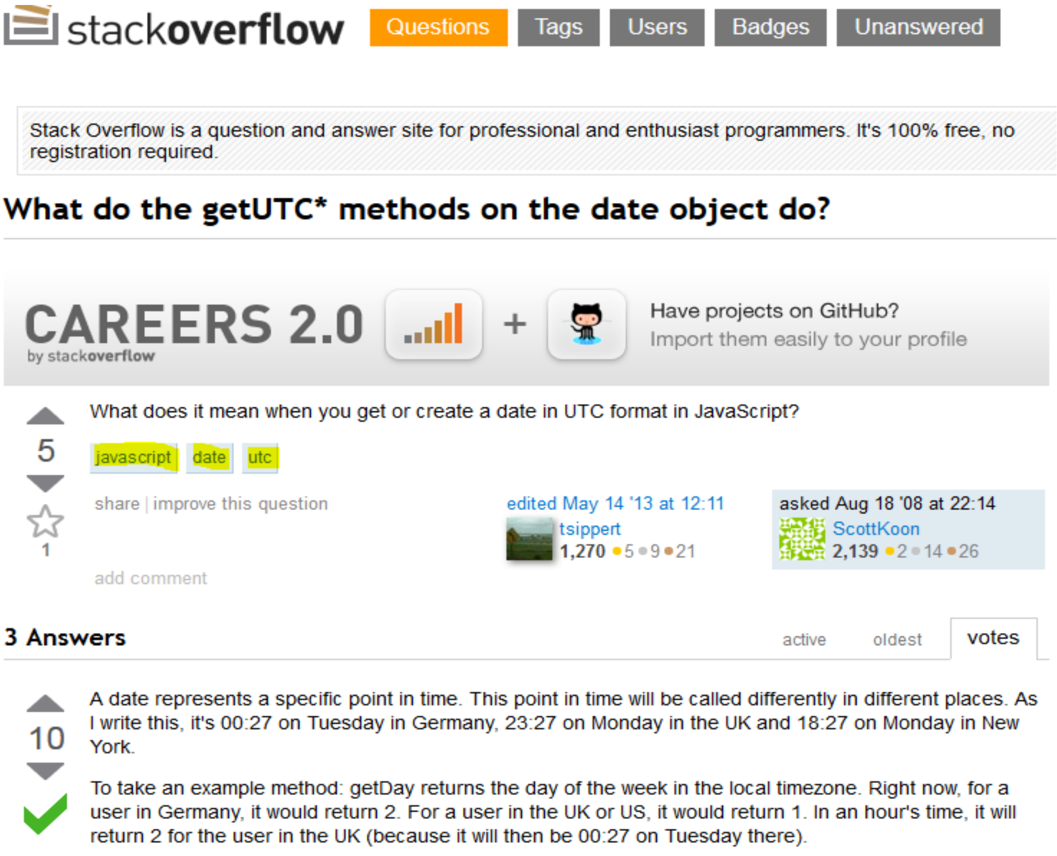
Generally, the expert users in CQA are more likely to provide satisfying answers under specific topics. Given a question, the problem is how to identify highly ranked authoritative answerers for askers. An early study on the expert finding [64] is to build user profile based on the questions they have previously answered, and then rank those user profiles according to the query questions. Another method in [120] proposed to study the link structure and topical similarity between askers and answers.

It is quite worthy to learn the expertise of users in expert recommendation for a given new question. Moreover, the user expertise can be used as a factor for ranking the answers to a question from different users. For expert learning, previous studies either neglected the user expertise or treat the voting scores of Q&A posts as the reflection of user expertise, such as [110]. However, the voting of questions have no direct relationship with user expertise. A user with high expertise under certain topics may have low descriptive ability to post a clear question, similarly, a user with high descriptive ability under certain topics may have poor expertise to provide relevant answers. Generally, a clear and readable question can receive more answers. [110] assumed that the user expertise is reflected by the voting scores of Q&A postings of a user. However, the voting scores of questions and answers offer different explanations. The former measure the clearity and usefulness of the questions, while the latter evaluate the usefulness of the answers to corresponding questions. Hence the expertise of users mainly depends on the voting information of answers rather than that of questions.

Another problem is that the tags in each question were ignored in most work, which tend to be more informative than user profiles or user interested topics. In each question, the tags provided by askers can be viewed as the most representative words of the focus and intent of the asker. Generally, the voting score of an answer can reflect the expertise of its provider towards the corresponding question, which also can be viewed as the reflection of expertise on the tags of this question.

Figure 1.1 is a sample of Q&A posts from the popular Stack Overflow site. Three tags for the question “What do the getUTC* methods on the date object do?” are ‘javascript’, ‘date’ and ‘utc’, which are more representative than the generic topic for a question.

Due to the importance of tags, we propose to learn the user expertise based on tags and the voting scores of answers in CQA. Unlike existing methods, we won’t use the content of Q&A posts. Specifically, we build user-tag expertise matrix to depict the tag-level expertise of users. We assume that a user with the highest expertise score towards the tags of a given question tends to be the best candidate to be recommended.



Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required.

What do the getUTC* methods on the date object do?

Have projects on GitHub?
Import them easily to your profile

5

1

What does it mean when you get or create a date in UTC format in JavaScript?

javascript date utc

edited May 14 '13 at 12:11
tsippert
1,270 ●5 ●9 ●21

asked Aug 18 '08 at 22:14
ScottKoon
2,139 ●2 ●14 ●26

3 Answers

10

✓

A date represents a specific point in time. This point in time will be called differently in different places. As I write this, it's 00:27 on Tuesday in Germany, 23:27 on Monday in the UK and 18:27 on Monday in New York.

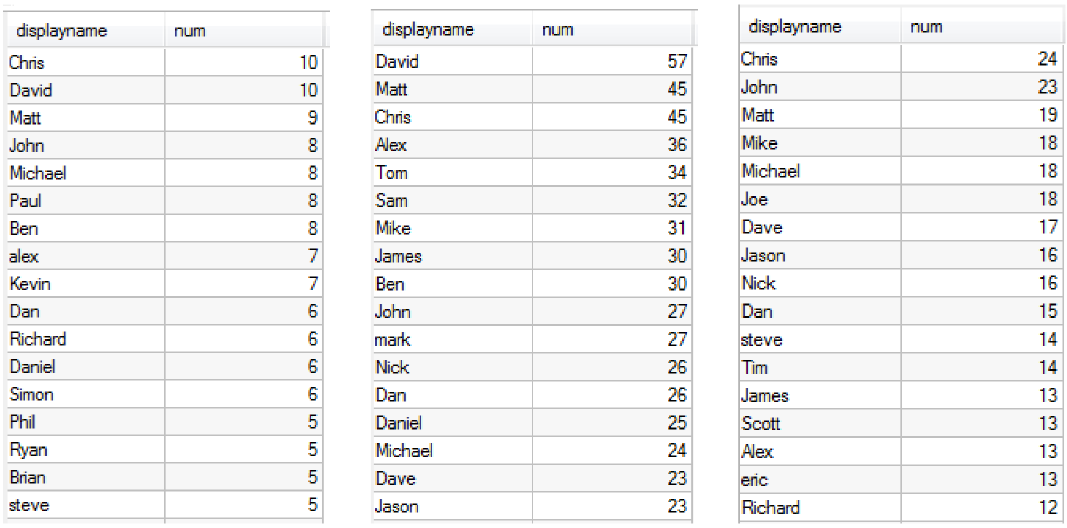
To take an example method: getDay returns the day of the week in the local timezone. Right now, for a user in Germany, it would return 2. For a user in the UK or US, it would return 1. In an hour's time, it will return 2 for the user in the UK (because it will then be 00:27 on Tuesday there).

Figure 1.1: An example of question-answer posts from Stack Overflow site.

It is not uncommon that some people in the CQA services share the same user names. Figure 1.2(a), Figure 1.2(b) and Figure 1.2(c) show three lists of user names from three different CQA communities: Travel[5], Webapps (Web Applications)[6], and Cooking[7], where each user name is shared by multiple users. In Figure 1.2(b), “David” is the most common and ambiguous user name related to 57 users. Figure 1.2(a) is based on the data between 2011-06-21 and 2013-05-09, Figure 1.2(b) is based on the data between 2009-07-15 and 2013-03-10, and Figure 1.2(c) is based on the data before 2013-03-10.

In some cases, an off-line person asks people around a difficult question orally, then he/she may be recommended by word of mouth to visit the CQA homepages of some potential answer providers. However, the links to their homepages are not provided sometimes, then the asker has to search them according to the provided user names. Some user names are unique, and

[5] http://travel.stackexchange.com/
[6] http://webapps.stackexchange.com/
[7] http://cooking.stackexchange.com/



displayname	num
Chris	10
David	10
Matt	9
John	8
Michael	8
Paul	8
Ben	8
alex	7
Kevin	7
Dan	6
Richard	6
Daniel	6
Simon	6
Phil	5
Ryan	5
Brian	5
steve	5

(a) Travel community

displayname	num
David	57
Matt	45
Chris	45
Alex	36
Tom	34
Sam	32
Mike	31
James	30
Ben	30
John	27
mark	27
Nick	26
Dan	26
Daniel	25
Michael	24
Dave	23
Jason	23

(b) Webapps community

displayname	num
Chris	24
John	23
Matt	19
Mike	18
Michael	18
Joe	18
Dave	17
Jason	16
Nick	16
Dan	15
steve	14
Tim	14
James	13
Scott	13
Alex	13
eric	13
Richard	12

(c) Cooking community

Figure 1.2: Example of lists of most ambiguous user names in some CQA communities (all the lists are not shown completely).

they can easily access the historical QA records of these potential answer providers. However, some are very common and ambiguous, accordingly, many users with the same user name will be displayed. Motivated by the above scenario, it is very necessary to help askers disambiguate these users, which can release them from wondering which user should be the right one. Moreover, if the user name is not clearly given, the askers will waste a lot of valuable time on searching and visiting irrelevant users, which can cause misunderstanding and misleading. Then the asker will get puzzled. Although there have been some studies on user name disambiguation in bibliographic citation records [26, 36, 100], the related methods are not directly applicable to our work, which is another motivation for our research.

1.2 Scope and Hypothesis

Although there are a variety of research directions in online social network analysis, the scope of this thesis mainly focuses on three areas: 1) Modelling users for community discovery in social networks. We assume that incorporating user sentiment-topic information into the model of community discovery enables us identify more interesting and specific communities. 2) Modelling expert users in question answering communities. We treat the voting scores of questions as the reflection of the descriptive ability of askers. We assume that a user who provides answers to an expert user is more likely to have high level ability as well. 3) Disambiguating users in question answering communities. This area is based on the assumption that the one who has the highest relevance score with a given question will be the right person to consider.

The overall hypothesis of this thesis is shown as follows.

We assume that modelling the user information, such as preference, sentiment, ability, is helpful for the community discovery and community question answering analysis. Specifically, as for community discovery, it is expected to find more representative communities, e.g, communities with obvious sentiment-topic distributions. When it comes to CQA, it is expected to improve the performance of expert user recommendation and the target user matching.

1.3 Main Goal and Contributions

The main goal of this thesis is to develop user information models to effectively address the above mentioned research problems in social networks.

1.3.1 User Social Sentiment-Topic Models for Community Discovery

Most of existing methods for community identification fail to consider the valuable sentiment factor in the networks. To directly detect the sentiment-topic level communities and to better explore the hidden knowledge within them, firstly, we propose a novel ***S**entiment-**T**opic model for **C**ommunity discovery*, called STC, to explore communities with different topic-sentiment distributions. The STC model is built by using social links, topics and sentiment in a unified way, where the sentiment is studied based on its corresponding topic. The main goal of this approach is to discover sentiment level communities, i.e., to find out some communities containing dominant sentiments on certain topics even though not all communities have dominant sentiment topics. In our model, we define a community as a collection of people who are directly or indirectly connected and share some sentiment topics with some members in this collection.

Experimental results on two types of real-world datasets demonstrate that our STC model can not only achieve comparable performance compared with a state-of-the-art community model, but also can identify communities with different topic-sentiment distributions, which might be applicable for the opinion analysis and decision making in large business and marketing service.

In STC model, the existence of users in each document is considered, while the author-recipient relationship in each document is ignored. To overcome this problem, we propose another two novel community models, one is an **A**uthor-based **S**entiment-**T**opic model for **C**ommunity discovery, called ASTC, the other is called ASTCx (the extension of the ASTC model). The main difference between them is that ASTCx depicts the sentiment words and topic words separately, while these words are mixed together in ASTC model. In each generative model, the social links, topics and sentiments are systematically combined. These three elements are very significant to the identification of meaningful community structures. However, it is not indicating that the more additional information incorporated into the model, the better result we can get. When the information is not important, the redundant factors can make the model more complex and inefficient. Note that not every community has sentiment information, our major focus is to get some communities including distinct sentiment polarities on certain topics. Experimental results and analysis on two real-world datasets show that our ASTC and ASTCx models can effectively uncover communities with different distributions. According to the distributions in communities, we can find sentiment unambiguous communities with respect to certain topics. Practically, those communities with distinct sentiment inclines are more attracting and valuable. In addition, our models have some real-world applications.

1.3.2 Expert User Recommendation in Community Question Answering

Existing studies considered that user expertise is reflected by the voting scores of both answers and questions. However, voting scores of questions are not really related to user expertise. We treat the voting scores of questions as the reflection of the descriptive ability of askers.

To better exploit the ability of users, we proposed to model both expertise and descriptive ability of users in question answering communities. Specifically, we present a novel probabilistic model, User Topical Ability Model (UTAM), to depict the topic-specific user ability, in which the textual information (words and tags) and voting scores of questions are combined to model the topical specific user descriptive ability, while the user topic-specific expertise is depicted by integrating the textual information (words and tags) and voting scores of answers. Apart from the intrinsic textual and voting information, we also explore the valuable social links within a QA community. To be exact, we proposed a new method, User Social Topic Ability (USTA), by integrating the results of UTAM with the link structure to further model the ability of users. The experiments conducted on the data from a very popular large CQA site, Stack Overflow, show that our models can perform competitively compared with the up-to-date methods in identifying experts, recommending suitable answers and retrieving similar questions.

The latest work considers to model the user expertise under topics, where each topic is learnt based on the content and tags of questions and answers. Practically, such topics are too general, whereas question tags can be more informative and valuable than the topic of each question. Due to the importance of tags, we view the tags of each question as the representative words of the subtopic of each question. In other words, we propose to learn the user expertise based on tags and the voting scores of answers in CQA. Unlike existing methods, we won't use the content of Q&A posts. Specifically, we build user-tag expertise matrix to depict the tag-level expertise of users.

Experimental analysis on a large data set from Stack Overflow demonstrates that our method performs better than the up-to-date method in expert user recommendation. Moreover, the lower time cost in the model training indicates that our method can be applied to very large datasets.

1.3.3 User Name Disambiguation in Community Question Answering

In CQA, to disambiguate the same-name users, we present a simple vector-style tag-based method, *relTagVec*, to learn the relevance between each user and the question by comparing their tag lists, where each tag is represented by a vector. Then the one who has the highest relevance score will be the right person to recommend. We also recommend the possible ranked user list when there are a great number of candidates. In addition, the title-based methods are introduced in evaluation. Experimental results on three CQA datasets from StackExchange[8] network demonstrate that our method is very effective, and performs much better than the baseline methods.

1.4 Thesis Structure

Chapter 2 gives the background and literature review related to the work in this thesis. The detailed contributions are described in Chapters 3 to 5, where the proposed approaches are evaluated and analyzed. Chapter 6 concludes this thesis and points out some weaknesses and suggestions. The organization of next chapters are briefly presented as follows.

Chapter 2

This chapter gives a detailed review on the relevant literature. First, we review the traditional and latest community detection methods, where the overlapping community detection methods are also introduced. Second, we describe the existing research topics and the relevant methods in community question answering. Then we provide a review about the algorithms for user name disambiguation.

Chapter 3

Chapter 3 is organized as follows: We first present our social sentiment-topic community discovery

[8] http://stackexchange.com/

model, STC, followed by the evaluation on real world datasets. We also give short discussion about STC model. To overcome the weakness of STC, we propose another two novel community discovery models, ASTC and ASTCx, then we report the experimental analysis and comparison for these models on real world networked data.

Chapter 4

In Chapter 4, firstly, we describe our UTAM model for expert user learning in community question answering. Then based on UTAM, we present our USTA method by incorporating social links. Thirdly, we propose a user-tag based model for expert recommendation in CQA. Finally, extensive experiments and comparison are conducted.

Chapter 5

Chapter 5 presents a novel vector-style tag-based method to disambiguate the same-name users in CQA. First, it shows the framework of our method, then it presents the evaluation based on two types of user names on the datasets from Stack Exchange network.

Chapter 6

In Chapter 6, we conclude this thesis. Specifically, we first summarize the contributions of this thesis, and then we point out the limitations in our work. Finally, we give suggestions and future directions based on these weaknesses.

Chapter 2

Background and Field Review

This chapter reviews the existing research work and background related to this thesis. Section 2.1 introduces various community discovery techniques for both static and dynamic networks. In Section 2.2, first, we present the preliminary knowledge on community question answering, then we introduce the main research areas and the corresponding methods in CQA. Section 2.3 introduces the problem of user name ambiguity, then it reviews the related work in user name disambiguation. Finally, we summarize this chapter.

2.1 Community Detection

In social networks, communities are considered as important structures in the form of groups of entities (people). There have been extensive studies on various community discovery problems [4, 18, 24, 31, 52, 62, 89].

2.1.1 Conventional Community Identification

In this section, we will introduce some traditional algorithms for community detection from networks. The methods for overlapping community detection and dynamic network evolution will be introduced in the following sections.

To start with an example, a small network divided into two communities by a dotted line is given in Fig.2.1.

An early significant approach, proposed in 2002 by Girvan and Newman, focuses on identifying community boundaries using centrality indices and edge betweenness [28]. Later, an algorithm using greedy optimization of modularity to infer community structure from network topology has been proposed by Clauset et al. [19]. As for a network with n vertices and m edges, the time com-

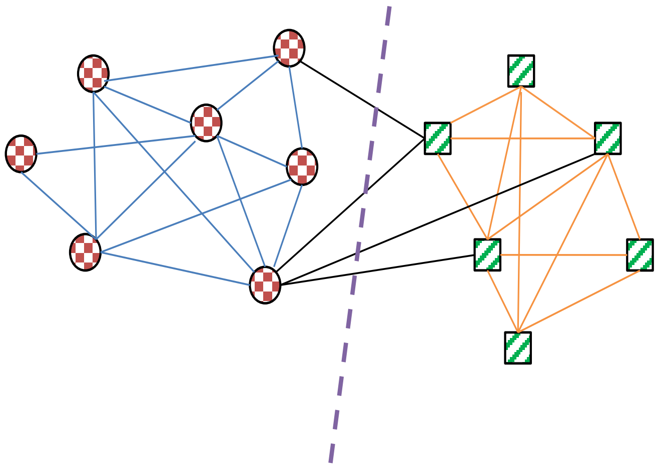


Figure 2.1: A partition of a small network into two communities indicated by a dotted line.

plexity of the proposed algorithm is $O(md\log n)$, where d is the depth of the dendrogram, $d{\sim}\log n$ when the network is hierarchical with roughly balanced dendrogram. For the sparse network, say, $m{\sim}n$, then the time complexity turns to be linear with $O(n\log^2 n)$. They apply this algorithm to a large network of items for sale on the Amazon online retailer website. The algorithm can identify clear communities within the network corresponding to specific topics or genres.

In recent years, several algorithms have been introduced to identify the known community structure in real networks, whereas these algorithms require users to provide the knowledge of the whole structure of the graph in advance. Practically, it is difficult to know fully about the large network like the World Wide Web.

To address this problem, Clauset [18] proposes a local modularity method and explores one vertex at a time in graph using a fast agglomerative algorithm because of lacking global knowledge. For general graphs, when there are n vertices being explored, and the average degree is d, then the time complexity of this algorithm is $O(n^2d)$. It indicates that the proposed algorithm can merge into web spider program to identify community structures in World Wide Web. In [71], Newman shows that the modularity can be rewritten in terms of the eigenvalues and eigenvectors of a modularity matrix for the network. The running time of this eigenvector-based spectral algorithm is mainly consumed for the evaluation of the leading eigenvector of the modularity matrix. It is demonstrated that this proposed algorithm with higher quality and running speed performs better than the state-of-the-art algorithms on a number of real-world network data sets.

Also, community structure identification in directed network has been paid increasing attention in the literature [2,33,74], whereas the majority work just apply algorithms devised for undirected networks, which ignores the useful information contained in edge directions.

To effectively find communities in directed networks, in [58], Leicht and Newman propose an extended well established modularity optimization method, which is on condition that a good partition will give a high value of the modularity Q. The real-world dataset used in the experiment is the network of American football teams competing in the Big Ten conference in 2005.

Although clustering in networks has been studied for many years, most general models based on exponential random graphs are not solvable for their properties. Later, Newman [72] proposed a random graph model of a clustered network, which is solvable for many of its properties, such as component size, percolation properties and giant component. This model could build an unbiased ensemble of networks with clustering to address the above mentioned problem in the network studies. It indicates that this model can also be extended to further studies about the impacts on processes like epidemic process.

Karrer and Newman [48] conduct an extension of the classical stochastic blockmodel [39] containing degree heterogeneity of vertices. A degree-corrected blockmodel is developed with closed form solutions, which performs much better in practice. When the heterogeneous degrees are incorporated, the performance of the models for statistical inference of community structure will accordingly be improved. However, the degree-corrected model also has its drawbacks, for the real-world networks, it may fail to represent high-order network structure accurately. Also, in fact, the number K of groups or blocks in the networks is unknown in advance.

2.1.2 Overlapping Community Detection

In this section, we show a review on the latest overlapping community detection algorithms.

Most of the existing algorithms are proposed for the disjoint community detection, such as modularity maximization, spectral clustering, and random walks. However, there usually exist overlaps among communities in the real-world networks. For example, a researcher can publish papers in several areas, like machine learning, database, and natural language processing.

The main notation is shown in Table 2.1.

CPM [80], a clique percolation algorithm, assumes that a community is composed of overlapping sets of complete subgraphs, so CPM is suitable for the detection in dense connected networks. An implementation of CPM is called CFinder[1]. An example of overlapping 4-clique-communities is shown in Fig.2.2, where community I overlaps with community II sharing two connected nodes.

[1] http://www.cfinder.org/

Table 2.1: Main notation for a graph.

Symbol	Description
$G = (V, E)$	a graph G with a set of n vertices V, and a set of m edges E
W	weight matrix
A	adjacency matrix
$C = \{c_1, c_2, ..., c_k\}$	A cover
$[a_{i1}, a_{i2}, ..., a_{ik}]$	belonging factor (soft assignment or membership)
a_{ic}	strength of association between node i and cluster c, $0 \leq a_{ic} \leq 1, \forall i \in V, \forall c \in C$, and $\sum_{c=1}^{\lvert C \rvert} a_{ic} = 1$

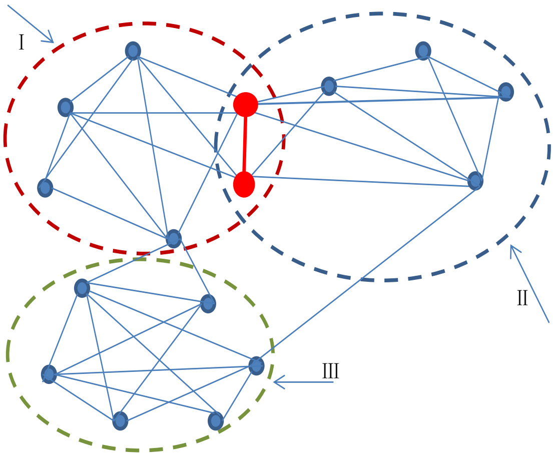


Figure 2.2: An example of overlapping 4-clique communities.

The overlapping nodes and link are emphasized, community III is not overlapped with I or II. Another clique-percolation-based algorithm called CPMw [24] is proposed for community detection in the weighted networks, where the intensity threshold for subgraphs is used to determine whether a k-clique should be included in a community or not. SCP [53] identifies clique communities of a predefined size unlike CPMw performing all values of k. Compared with CPM, SCP requires less computation time. However, the above mentioned clique-percolation-based methods are too specific, which are difficult to adapt to the real-world networks.

The link-based partitioning methods [1, 22, 23, 51] aim at partitioning links instead of nodes to identify overlapping communities. If a node is overlapping, then the links from this node should belong to more than one group. Actually, grouping edges and grouping vertices are somewhat symmetric [27]. A weighted line graph framework for overlapping community detection is described by Evans and Lambiotte [22, 23], where the links of the original graph can be viewed as the nodes of the line graph. Accordingly, the edge partition of the original graph is the partition

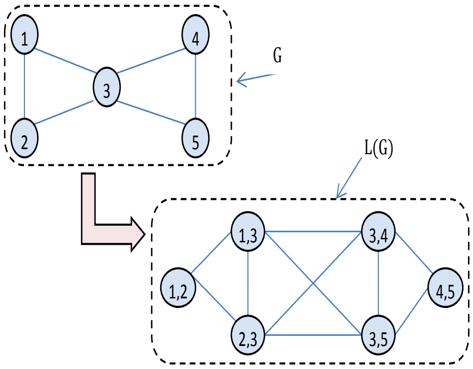


Figure 2.3: The transformation from a graph G to its line graph $L(G)$.

of nodes of a line graph. Fig.2.3 gives an example of a line graph $L(G)$ and its corresponding original graph G. In this line graph $L(G)$, each node is associated with the endpoints of the edge in the original graph. Recently, Evans [21] has extended the line graph model to a clique graph.

Fuzzy methods are also very popular in overlapping community detection, such as fuzzy clustering based technique [117], vertex similarity based approach [70], and model-based probabilistic algorithms [56, 67, 87]. In these algorithms, a fuzzy (soft) membership (assignment) vector is computed for each node. As the notation in Table 2.1, given a node i and the community set $C = \{c_1, c_2, \ldots, c_k\}$, then the corresponding membership vector of node i can be written as $[a_{i1}, a_{i2}, ..., a_{ik}]$, where $0 \leq a_{ic} \leq 1$ for each $c = 1, 2, \ldots, k$ and $\sum_{c=1}^{k} a_{ic} = 1$ for each $i = 1, 2, ..., n$. The number of communities is required to provide in advance, which is a disadvantage of such methods. There is an example demonstrated in Fig.2.4, where the nodes and their fuzzy memberships are figured out, and the final partition of these nodes are determined when the threshold (e.g., 0.3) is given. It is obvious from Fig.2.4 that node 2 belongs to two communities.

In the local based methods, the local expansion and optimization are used to help the detection of communities. In [5, 6, 38, 46, 49, 54, 55, 57, 75], algorithms are based on finding the local optimization of different fitness functions. These functions are used to measure the quality of a community. For example, in [54], the fitness function is $f(c, \beta) = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\beta}$, where k_{in}^c and k_{out}^c are the total internal and external degree of the community c, and β is a controlling parameter. An example of a community discovered by a local method is shown in Fig.2.5, the gray node in the center is considered as a seed community, and then the community is expanded by the black nodes with positive fitness values, while the red nodes are not included in this community due to the negative fitness values.

An earlier local-based method for overlapping community is described by Baumes et al. [5,6].

node	Community 1	Community 2	Community 3
1	0.031	0.927	0.042
2	0.441	0.101	0.458
3	0.073	0.148	0.779
4	0.984	0.009	0.007

Partition results, threshold=0.3

Fuzzy membership

node	Community 1	Community 2	Community 3
1	0	1	0
2	1	0	1
3	0	0	1
4	1	0	0

Figure 2.4: An instance of the nodes, the corresponding fuzzy memberships (belonging coefficients), and the partition results.

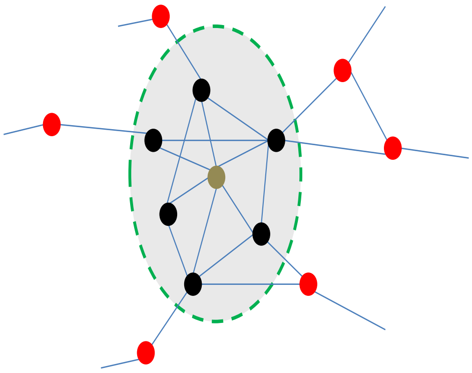


Figure 2.5: A community identified by a local method.

This method contains two parts: 1) Ranking nodes using RankRemoval algorithm, then removing highly ranked nodes from the network iteratively until small to form the cluster cores; 2) Iterative Scan, the cores are viewed as seed communities and would be expanded by updating nodes until a local density function converges.

A novel algorithm called COPRA (Community Overlap PRopagation Algorithm) [31] is introduced by Gregory on the basis of the label propagation technique [84]. COPRA allows a node label to include multiple community identifiers, and the label of a node at each time step is updated according to its neighbors' labels. For each iteration, the time complexity is $O(\xi m \log(\xi m/n))$, where ξ is a parameter influencing the maximum number of communities to which a node belongs. Fig.2.6 gives an example for label propagation, the maximum number of communities

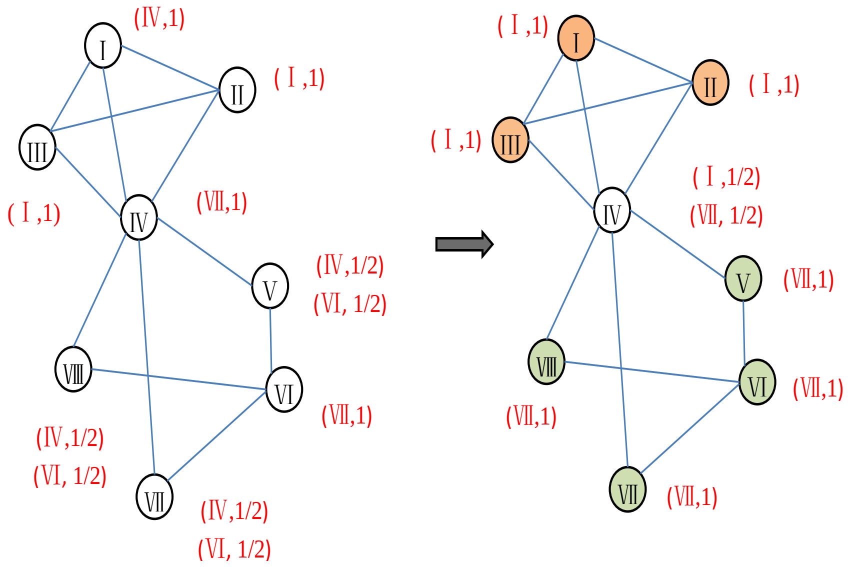


Figure 2.6: An example of propagation of labels, the maximum number of communities that a node belongs to is 2. Here, node IV belongs to two communities.

that a node belongs to is 2, finally two overlapping communities are obtained: $\{$I,II,III,IV$\}$ and $\{$IV,V,VI,VII,VIII$\}$. Another propagation-based algorithm, SLPA (Speaker-listener Label Propagation Algorithm) [105, 106], propagates labels by simulating the human pairwise interactions, which is a new fast detection algorithm. In SLPA, each node is given a memory to keep labels. It is different from the algorithms in [31, 84], where the knowledge received from previous iterations is not kept in node.

Apart from the above classes of algorithms, some other algorithms are also presented for overlapping community detection. GONGA [29] is proposed to identify overlapping communities in networks, which extends the GN (Girvan and Newman) algorithm [28] by spitting vertices. The edge betweenness, vertex betweenness, and split betweenness are calculated in the GONGA algorithm, and the worst-case time complexity is $O(m^3)$, m is the number of edges. To increase the speed of GONGA algorithm, Gregory [30] introduces an improved algorithm GONGO (GONGA Optimized) using a local betweenness. The time complexity of GONGO for sparse network is $O(n \log n)$. In [86], Rees and Gallagher describe an algorithm to detect overlapping communities from egonets based on group viewpoint of individuals. This algorithm contains two stages: 1) Identifying friendship groups; 2) Aggregating friendship groups into communities.

2.1.3 Community Detection in Dynamic Networks

In recent years, the research on community detection in dynamic networks has been paid increasing attention [3, 13, 17, 25, 40, 50, 62, 79, 95, 97]. We give an example of the evolution of a network changing with time in Fig.2.7, the interaction between nodes can emerge and disappear in different timestamps.

An early work for studying evolving communities in dynamic networks is described in [40], Hopcroft et al. track the evolution of communities over time, where communities are identified by an agglomerative clustering. The subsets of clusters that keep stable and suffer little from slight perturbations of graph under a series of clustering runs are viewed as natural communities, which help us to track the temporal evolution and detect emerging new communities. Two snapshots of a real-world network are used in the experiments.

To identify community evolution, a novel algorithm based on clique percolation method is p-resented in [79]. They illustrate the basic events occurring in the lifetime of a community: birth of a new community; growth or contraction; merge or split; and the death. For a target community, the size s and the age τ (the time since its birth) are generally considered. An auto-correlation function $C(\cdot)$ is introduced to measure the relative overlap between two states of the same community. It is demonstrated that the value of $C(\cdot)$ declines faster for the larger communities, which indicates that the larger communities are more dynamic.

To study the evolution of interaction graphs, Asur et al. [3] introduce an event-based framework. An evolving network can be converted into static snapshot graphs, then clusters are obtained using the MCL algorithm [20] at each snapshot separately. Critical events are identified to offer novel information for the learning of dynamic behavior of networks. Two kinds of critical events are described: 1) events involving communities (i.e., continue, k-merge, k-split, form and dissolve); 2) events involving individuals (i.e., appear, disappear, join and leave). The behavioral tendencies of nodes are related to the evolution of the network. Four novel behavioral measures, i.e., stability index, sociability index, popularity index and influence index are introduced based on the critical events, which can be used for link-prediction and influence maximization. The experiments are conducted on two different dynamic networks, DBLP collaboration network and Clinical Trials dataset.

GraphScope [95] is proposed for the mining of large-scale evolving networks, which is focused on two major problems: 1) community identification, and 2) change detection in a graph stream. Moreover, GraphScope is a parameter-free and automatical approach based on the principle of Minimum Description Language. They consider the problem of the evolving bipartite

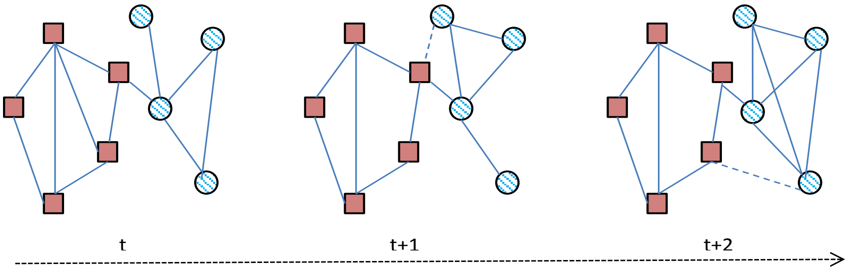


Figure 2.7: A network evolves over time (snapshot graphs).

graphs, wherein the source and destination nodes are treated independently. GraphScope aims at identifying the number and location of the change-points, as well as the number and membership of the communities under minimal cost. It is demonstrated that GraphScope can uncover meaningful patterns from evolving networks, also it is very efficient, effective and scalable. However, it fails to track the evolution of individual community.

The traditional methods for learning community evolution [3, 79, 95] are mainly divided into two steps: 1) identifying communities independently at each timestamp; 2) extracting community evolution over time. However, the real-world data is noisy. In this case, the traditional methods usually produce high temporal variation between communities.

To address this problem, Lin et al. [62] present a FacetNet framework in which communities and their evolutions are studied in a unified process. Then an iterative algorithm is used to get an optimal solution. At a given timestamp, the network data and the historic evolution patterns are combined to determine the community structure. This framework makes the produced communities and evolutions less sensitive to noise. However, due to the high cost of computation, FacetNet is not scalable to large-scale networks.

Also in [50], Kim and Han introduce a novel evolutionary clustering technique. The concepts of nano-communities and quasi l-clique-by-clique are presented, which help to identify a variable number of communities. An information-theory based mapping algorithm is introduced for the evolving, forming and dissolving of each community. Experimental results demonstrate that the proposed method can perform better than FacetNet framework [62] in terms of clustering accuracy and running time.

2.1.4 Community Detection Using Link and Content

The majority of existing community identification methods ignore the content of social interactions in social networks.

An early framework for community discovery using link and content elements is proposed in [119], the authors propose two community-user-topic (CUT) models based on joint user and topic distributions for semantic community discovery. The first generative model, CUT_1, is modelling community with users, where each community is viewed as a distribution over users, each user is associated with some topics, and each topic follows a distribution over words. For model CUT_1, the joint distribution of community c, user u and topic z, and word w is $p(c, u, z, w) = p(w|z)p(z|u)p(u|c)p(c)$. Unlike CUT_1, the second generative model, CUT_2, is to model community with topics. Specifically, the relations between community and topics are mainly emphasized, users who have few communications but share common topics can also belong to the same community. In the CUT_1 model, the tie between community and topics is loose, where the topics are directly conditioned on user but not community. The joint distribution of variables for model CUT_2 is $p(c, u, z, w) = p(w|u)p(u|z)p(z|c)p(c)$. For the inference and parameter estimation in the two models, an EnF-Gibbs sampling algorithm is proposed by integrating Gibbs sampling and entropy filtering, by which the non-informative samples are automatically filtered. To evaluate the performance of the proposed two models, an existing topology-based baseline algorithm is used for comparison, which is also a Modularity based community detection method. Experimental evaluations are conducted on the popular Enron email dataset. To compute the similarity between the communities discovered by the two CUT models and the baseline algorithm, a clustering comparison method [85] is used. The experimental results show that the proposed models can effectively detect communities. In addition, the authors present that the similarity value between CUT_1 and Modularity is larger than that between CUT_2 and Modularity. In the two CUT models, user and topic are considered for community discovery. However, the two factors are not well integrated.

CART (Community-Author-Recipient-Topic), another Bayesian generative model [81], is proposed to integrate link and content information in the social network for discovering communities, which is an extension of the Author-Recipient-Topic (ART) model [66]. It is assumed that the authors and recipients are generated from a latent group, and actors can belong to multiple communities, whereas each document is assigned to one community. Figure 2.8 shows the graphical representation of the CART model, where β, α and η are the Dirichlet priors, λ is topic distribution over words, θ denotes the topic mixture, and ϕ is the user mixture. The observable variables are author a, a set of recipients b, and words w. In addition, the community assignment c, recipient r and topic z are latent variables. M is the number of communities, N denotes the number of word tokens in an email document, K is the total number of topics, R is the number of recipients for

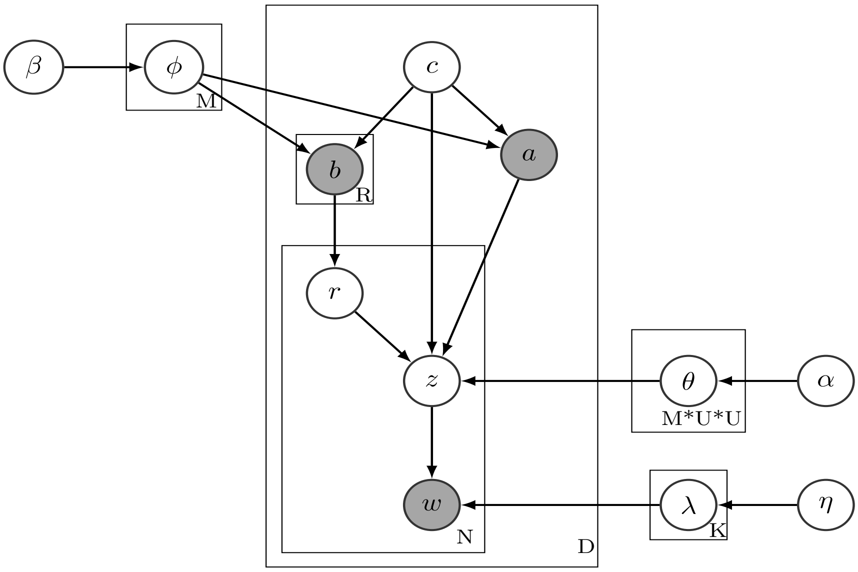


Figure 2.8: Graphical representation for the CART model.

an email, and U is the total number of users. The generative process for each email document d $d = 1, 2, ..., D$ in CART model can be described as follows: 1) Sample a community c_d randomly, which has a uniform distribution. 2) Choose the author a_d and the corresponding recipients b_d. 3) To generate the i-th word in the email d, sample a recipient $r_{d,i}$ uniformly from b_d. 4) Draw a topic $z_{d,i}$ from the topic mixture $\theta_{c_d,a_d,r_{d,i}}$. 5) Then sample a word $w_{d,i}$ based on the $z_{d,i}$-th topic-word distribution. For the model inference, a Gibbs sampling algorithm is employed. Experiments on the Enron email corpus demonstrate that the proposed model can discover meaningful communities and related topics. The authors list the top words for each social topic learned by the CART model. For the community evaluation, the topic probabilities and actor profiles for communities are illustrated respectively.

In [111], Yang et al. propose to integrate a popularity-based conditional link model (PCL) with a discriminative content (DC) model into a unified framework to discover communities. In PCL model, the main task is to model $P(j|i)$, the probability of linkage from node i to node j among all the nodes in $\mathcal{LO}(i)$, here $\mathcal{LO}(i)$ is the link-out space of node i. The content information being incorporated into PCL is utilized to model the memberships of nodes via a discriminative approach. For maximum likelihood inference of the combined model, a novel two-stage optimization algorithm is proposed. Two variants of this framework by combining link and content information are also presented, the first one is **PCL+PLSA**, where the PCL model is combined with PLSA model, another one is named as **PHITS+DC**, here PHITS model is used for link analysis. All the two variants are used as baseline models for comparison. In the experiments, four datasets are used: 1) Political blog data set; 2) Wikipedia data set; 3) Cora data set; 4) Citeseer dataset. To measure the performance on link prediction, the popular *Recall* metric is used in the comparison between PCL model and PHITS model. For the evaluation on community detection, four metrics are used: 1) normalized mutual information (NMI); 2) pairwise F-measure(PWF); 3) modularity(Modu); 4) normalized cut (NCut). The experimental results show that the combined model performs better than the existing methods. However, this model ignores the topics in the networked data.

Another novel method for detecting communities in social networks using interactions and content is proposed in [89]. In such method, the discussed topics, social links, and interaction types are all used to build several generative community models, namely, TUCM (Topic User Community Model), TURCM-1 and TURCM-2 (Topic User Recipient Community Models) and full TURCM model. For the networks like Twitter, the message broadcasts are common, and the topics of the posts are mainly related to the senders' interests. In such case, the authors proposed the TUCM community discovery model, which is built based on the user generated content and the type of their interactions. It is assumed that a user can discuss multiple topics and can be the member of multiple communities. The number of communities C and the number of topics K are apriori. The graphical notation of TUCM model is shown in Figure 2.9. In Figure 2.9, ξ, δ, α and β are the hyper-parameters, η is the topic mixture, π denotes the topic distribution over words, θ is the community proportion, and ϕ represents the interaction type mixture. The observable variables are user u, word w and the type of posts x. The topic z and community c are the latent variables. In addition, U, K and C are the number of users, topics and communities. P_i denotes the posts sent by user u_i, and N_p is the number of word tokens in post p. The joint probability distribution for the TUCM model can be represented as Eq.2.1.

$$\begin{aligned} &P(\mathbf{w}, \mathbf{c}, \mathbf{z}, \mathbf{x}, \mathbf{u}, \eta, \theta, \pi, \phi | \delta, \alpha, \xi, \beta) \\ &= P(\mathbf{z}|\mathbf{u}, \eta) P(\mathbf{c}|\mathbf{z}, \mathbf{u}, \theta) P(\mathbf{w}|\mathbf{z}, \pi) P(\mathbf{x}|\mathbf{c}, \phi) P(\pi|\delta) P(\theta|\alpha) P(\eta|\xi) P(\phi|\beta) \end{aligned} \tag{2.1}$$

For another kind of networks (like email networks) with little mass messaging, where the sender-recipient pairs of posts are very common, and the post topic is related to the interest of both senders and recipients, the authors proposed three TURCM models. Here we introduce the full TURCM model. Figure 2.10 shows the graphical notation of the full TURCM model, where ε is also a hyper-parameter, the recipient r is an observable variable, which is not described in TUCM model, and ψ is the social recipient mixture. R_p denotes the number of recipients of post p.

To infer the parameters for TUCM model and the TURCM models, a Gibbs sampling based

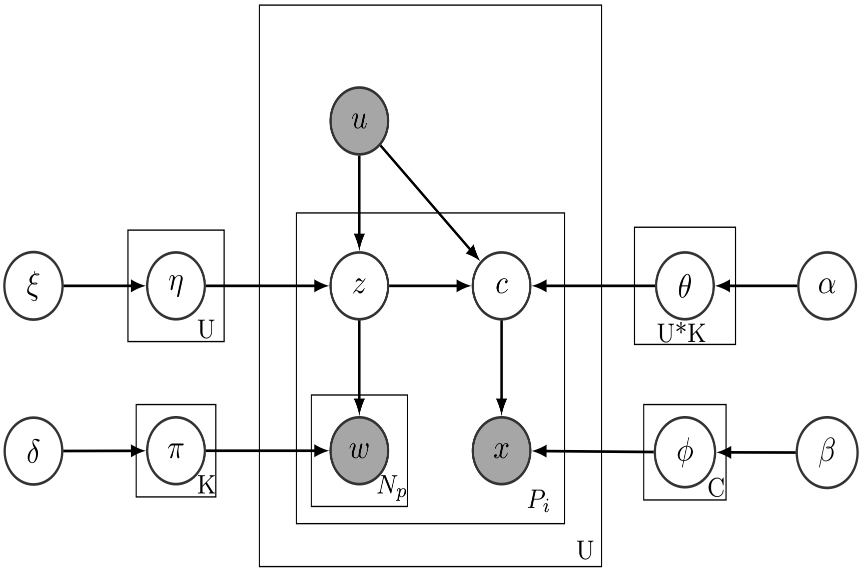


Figure 2.9: Graphical representation for the TUCM model.

algorithm is used. The evaluations are conducted on two different social network datasets, one is a twitter dataset, the other is from the Enron Email corpus. In the qualitative analysis, the authors reported: 1) The top words for each topic and user roles; 2) The topic and community proportions for a user; 3) The distribution of topics within communities. To measure the performance of community discovery, the aforementioned CUT [119] and CART [81] models are used as baselines. A metric named fuzzy modularity is introduced. The results demonstrate that the full TURCM model discovers most meaningful communities. In addition, the authors conducted the perplexity analysis, where both words and link types are considered. In the time analysis, it shows that the TUCM model is more efficient.

More recently, a community profiling model, COCOMP (COllaborator COMmunity Profiling), has been proposed by Zhou et al. in [125] to identify the communities of each user and their relevant topics and groups. The social links and topics between users are both considered in COCOMP. The graphical notation of COCOMP model is shown in Figure 2.11. In Figure 2.11, the topic proportion θ has a Dirichlet distribution with prior α. The participant mixture η has a Beta distribution with priors α_0 and β_0. ϕ is the topic-word distribution with prior β. ψ denotes the community proportion with hyper-parameter μ. With regard to the variables, topic z and community c are latent, the word w and user participation i are observable. Here D is the number of documents, N is the number of word tokens in a document, K, M and P are the number of topics, communities and users respectively. The joint probability for COCOMP model can be written as

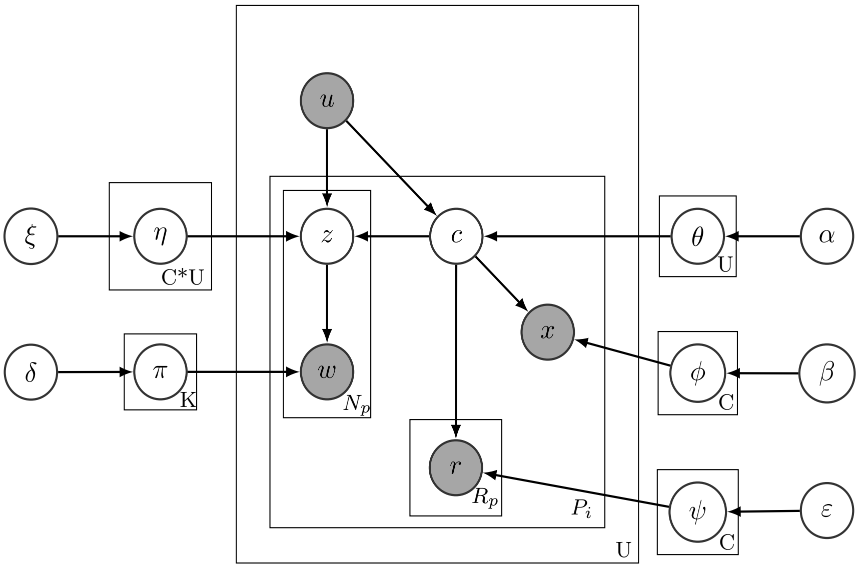


Figure 2.10: Graphical representation for the full TURCM model.

Eq.2.2. For the model inference and parameter estimation, a Gibbs sampling method is employed. The experiments are conducted on several email and twitter datasets, and the metric for evaluation is perplexity. For community analysis, the communities' topic distributions, the main topics and people in some users' communities are shown, where each topic is described by some keywords. For comparison analysis, LDA model is used as a baseline. It is demonstrated that COCOMP model can discover meaningful communities, and can perform better than LDA.

$$\begin{aligned} &P(\mathbf{c}, \mathbf{i}, \mathbf{z}, \mathbf{w}, \eta, \psi, \theta, \phi | \alpha, \beta, \mu, \alpha_0, \beta_0) \\ &= P(\mathbf{c}|\psi)P(\mathbf{i}|\mathbf{c}, \eta)P(\mathbf{z}|\mathbf{c}, \theta)P(\mathbf{w}|\mathbf{z}, \phi) \\ &\quad P(\eta|\alpha_0, \beta_0)P(\psi|\mu)P(\theta|\alpha)P(\phi|\beta). \end{aligned} \tag{2.2}$$

However, the above mentioned methods fail to consider the sentiment information of topics, which is an important factor when discovering more meaningful communities.

2.2 Community Question Answering (CQA)

Community question answering (CQA) has attracted much attention in recent years, which is a popular type of question answering service. Unlike traditional QA system, the main goal of CQA sites is to establish online platform for us to ask questions or answer the questions from others.

In the past years, research on CQA has achieved great progress in various aspects, such as expert user recommendation [44, 122], answer recommendation [121], similar question retrieval

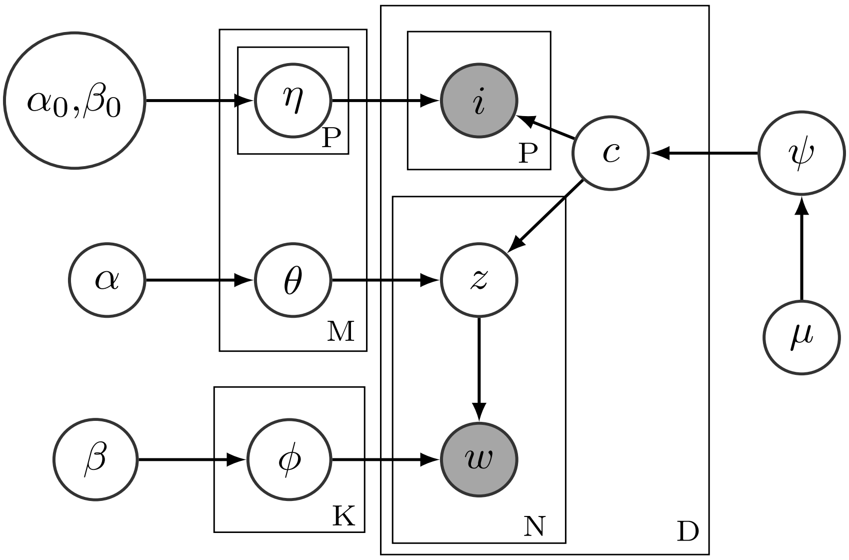


Figure 2.11: Graphical representation for the COCOMP model.

[12, 116], question subjectivity prediction [59, 59, 124], question classification [10, 93], answer summarization [14, 99], and user intent discovery [112, 113], etc. In this section, we mainly introduce the first three aspects due to the high relevant to our work.

2.2.1 Expert User Identification in CQA

Expert user learning is an important topic in CQA. In [64], the experts are discovered by combining user profiles obtained from the previously answered questions and up-to-date information retrieval methods. The authors treat the expert discovery as an information retrieval problem, and the goal is to rank the user profiles given the query question, then the users who has the higher ranking are more likely to be selected. In [47], the popular HITS algorithm was extended for finding experts in CQA, and the link structure of community was well studied. The experiments on Yahoo! Answers dataset show the effectiveness of the method. In [115], Zhang et al. proposed to find authoritative users via network-based ranking algorithms in social network, the authors observe that the performance is close to the human raters, in addition, the structural feature of network is very important. Bouguessa et al. [9] proposed a new method to identify authoritative users in CQA, where the user authority scores are presented as a mixture of gamma distributions, and the number of authoritative actors is determined automatically, where the EM algorithm is used in the parameter estimation. Experimental results on a large Yahoo! Answers dataset demonstrate that the proposed method can automatically identify authoritative users. Moreover, it indicates that the user generated content is important.

In [34], a two-step approach was proposed for answer provider recommendation, in which the user profiles are created based on latent topics and users latent interests, then the answerers for a new question are determined via term-level information and topics of users and questions. That's to say both the term-level and topic-level information are integrated in the research. The authors propose a simplified User-Question-Answer (UQA) model by assuming that the topic spaces for both the question and answer content are the same. Experimental results show that the combination of the two kinds of information can make improvements. In [8], a semi-supervised coupled mutual reinforcement framework was proposed for computing quality of content and user reputation scores, where the question and answer quality and user reputation is considered. The experimental results demonstrate that the proposed semi-supervised method can improve the performance of the supervised method.

A novel question selection bias method [78] was proposed for user behaviour study, which can identify experts by Gaussian classification models. The authors assume that the experts would like to questions which have not receive good answers, another assumption is that the valuable question is more likely to be noticed by the expert users. Moreover, the authors also present that the combination of bias and other simple measures can further improve the performance.

Liu et al. [63] proposed to estimate the expertise scores based on competition between each pair of users, where the two competition-based methods, TS and SVM, for expert score estimation. The authors view the pairwise comparison as the two-player competition. Experimental analysis on NTCIR-8 CQA data proves that the competition-based method performs better than link-analysis based and pointwise based approaches. In [15], a new bias-smoothed tensor model was proposed for user reputation score estimation under a comment rating scenario, and a novel latent factor model is used for rating prediction, where the support-based reputation score is considered. Zhou et al. in [120] proposed to integrate both the link analysis and topical similarity to identify experts in CQA by a probabilistic model, namely, a topic-sensitive random surfer model (TSPR). In this model, the topics are learned based on the historical questions and answers of users. In addition, the expert saliency score is measured based link structure and topical similarity. The experimental results on Yahoo! Answers show that the proposed method performs better than the link-based methods. In [77], question selection bias is used to discovering current and potential experts. The authors consider not only the problem of finding the current experts, but also the future potential expert users. Experiments demonstrate that the question selection bias can improve the performance of baselines.

More recently, a novel method for identifying potential contributive users by exploiting their productive vocabulary is presented in [96]. To measure the user contributive likelihood, the expertise and availability of each candidate user are mainly considered. Another new method considering the cold-start expert identifying in CQA is proposed in [118], firstly, the authors create the user-user social graph by using the user links and topical interests, then to predict the user expertise, a graph regularized latent model is introduced, where both user previous question answering records and the social graph are used in the modelling.

2.2.2 Relevant Answer Identification in CQA

Given a new question, there have been some methods for ranking candidate answers. Berger [7] proposed to study the correlation between questions and answers for answer finding, where the lexical chasm between questions and answers is connected, moreover, the translation model is used. In [43], the non-textual features were concerned in the prediction of the quality of answers, in which the maximum entropy approach were used, where the quality of documents are predicted via a stochastic process, and a number of features like answer length, answerer's acceptance ratio are considered. Most existing research studies for answer ranking is based on machine learning methods and features.

Recently, Zhou et al. [126] studied the relationship between rich profile information and answer quality for answer ranking in community question answering, where three types of user profile information is considered, namely, engagement-related, authority-related, and level-related. Experimental analysis on Yahoo! Answers demonstrates that the importance of the user profile information, especially the engagement-related information. Another approach [98] was proposed to identify best answers based on the key features extracted from the labelled data, the authors build classifiers for the prediction of best answers based on the labelled data. However, the real-world data is generally unlabeled.

2.2.3 Similar Question Retrieval in CQA

When it comes to the similar question retrieval, Jeon proposed to find similar questions based on the similarity between answers in QA archives [42], where the translation-based retrieval model performs better than other methods. In [104], a probabilistic latent semantic analysis (PLSA) based incremental recommendation algorithm was proposed, where the long-term and short-term user preference, negative and positive user feedback are taken into account. Experimental studies show that the proposed method is flexible and superior to the baseline methods.

And in [83], the classical PLSA model was used in similar question retrieval, where the user interest is learned based on their previous questions. The evaluation on Yahoo! Answers data demonstrates the effectiveness. In [11], to retrieve similar questions, the semantic similarity based latent topics and the translation based language model are integrated, where the category information of questions are considered. Recently in [45], a Question-Answer Topic Model (QATM) is learned to reduce the lexical gap between the old questions and new queries for question retrieval, it is based on the assumption that each question-answer pair share the same topic distribution. To improve the performance, the authors extend QATM with posterior regularization and show that the proposed models perform better.

Another approach [37] using the question patterns generated from previous questions to identify more equivalent patterns via a novel bootstrapping-based method. Zhou et al. [121] utilize the useful cross-language semantic information to tackle the problems of word ambiguity and mismatch in question retrieval task. To enhance the question retrieval, statistical machine translation and matrix factorization techniques were employed. In [82], a unified model called LSTI was proposed to retrieve similar questions in CQA, which combines latent semantic indexing and tensor analysis to model word associations. Zhou et al. [123] created a semantic relation based concept thesaurus from Wikipedia knowledge, which is then used to improve the performance of question retrieval in concept space.

Recently, a topic expertise model (TEM) for community question answering services was proposed [110], in which the user topics and topical expertise are jointly studied. To exploit the social link information between users, a CQARank method was presented based on the results of TEM and link structure analysis, which can further improve the performance of TEM for expert learning, answer recommendation and similar question retrieval tasks.

However, existing methods ignored to model the user descriptive ability, which can be learnt from the voting of user posted questions. Although in [110] the voting scores of question are also studied, whereas which were treated as the reflection of user expertise.

2.3 User Name Disambiguation

Name ambiguity has been a very popular problem in different domains, like in publications or bibliographies, which is a particular case of identity uncertainty. It is very common that one person can share the same user name with other people, which make the name ambiguous. To address name ambiguity, a variety of methods have been proposed for user name disambiguation, such as [26, 36, 100, 114].

In an early work [65], the authors propose to identify target users with ambiguous user names via feature extraction and unsupervised clustering methods, where each text document is assigned a feature vector, and a bottom-up centroid agglomerative clustering method is used to merge similar vectors.

In [35], Han et al. use two supervised methods, i.e., Naive Bayes and Support Vector Machines (SVMs) to author disambiguation in users' citations. Additionally, several features like co-author names are used in the methods. However, these supervised methods are not suitable for large scale digital libraries. The reason for this is that the manual labeling or annotation of the training data is time consuming. Next, [36] presents an unsupervised approach to disambiguate users in bibliographies by employing a K-way spectral clustering method, where three kinds of features are utilized, namely, paper title, co-author names, and publication venue names. The authors create citation vectors for each name dataset, then the multi-dimensional citation matrix is built. Experimental results on various name datasets indicate that the spectral based methods can yield better performance. In addition, the authors show that these features can improve the accuracy of disambiguation. Another insight is that a number of factors like dataset size, the author research area diversity can influence the performance.

In adition, in [94], Song et al. propose a topic-based framework for user name disambiguation in publications and web pages. Specifically, at first, two topic models are built based on Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (PLSA), then a hierarchical agglomerative clustering method is employed on the topic distributions. Pair-level pairwise F1 score and cluster-level pairwise F1 are used as the evaluation metrics. The authors show that the proposed approach performs better than several unsupervised clustering methods.

For user name disambiguation in citations, [109] explores the relationship between citations to check if the same user name from two citations belongs to the same author, where topic correlation (measuring the topic similarity) and web correlation (measuring the co-occurrence times) is calculated. Then the citations are grouped into clusters via a pair-wise grouping method.

In [100], a random forests based machine learning algorithm is introduced for pairwise user name disambiguation. In addition, 21 similarity profile features are defined and grouped into six types, where the similarities are defined based on author, affiliation, journal, title, coauthors and concept. The experiments show that the random forest model performs better than SVM method for the pairwise user name disambiguation in academic publications.

A novel approach, Self-training Associative Name Disambiguator [26], is proposed for author name disambiguation through two steps. Firstly, a clustering method is used to automatically get examples, which can reduce the cost of manual labeling. Secondly, an associative name disambiguator is employed to identify unseen users who are not in the training set. The experimental analysis on datasets from two digital libraries show that the proposed disambiguator performs better than the unsupervised methods and comparable with the supervised methods, the main advantage is that it can obtain training example automatically.

In [102], the authors propose a novel active name disambiguation method, where the active user interactions are considered. Firstly, a pairwise factor graph (PFG) model is created, then a number of features, like citation, coauthor, are integrated with PFG model. The evaluation shows that the user corrections can improve the disambiguation performance.

Another work [16] presents a bootstrapping name disambiguation algorithm. At first, a conditional pair-wise graph model is built, then the training examples are obtained via an newly proposed active data augmentation method, and then the disambiguation is conducted by using a graph partition based method.

Recently, a new method has been presented in [114] by exploring the link information in collaboration social networks for disambiguating user names, where only the graph structure information is used for disambiguation without intruding the privacy of users.

2.4 Summary

In this chapter, we provide a detailed review about the related literature to our research in this thesis. First, we introduce the existing community detection methods. Second, we present the existing methods for several research problems including expert user recommendation, similar question retrieval in community question answering. Then we review the algorithms for user name disambiguation in different networks, which is also related to our research work.

Chapter 3

Modelling User Sentiment-Topics and Links for Community Discovery

Social networking services are attracting increasing interest in the domain of community discovery. In social networks, the interactions among users are very frequent by sending emails, posting tweets, and sharing comments online, etc. Such networks usually include rich sentiment information, which can provide us with useful resources for identifying communities with different sentiment-topic distributions. Traditional methods for identifying communities in networks are based on direct link structures, which ignore the content information shared among groups of entities. Recently, community detection approaches by using both link and content have been studied. However, most of these methods are not capable of identifying sentiment-topic based communities. In this chapter, to directly detect the sentiment-topic level communities and to better explore the hidden knowledge within them, at first we propose a novel **S**entiment-**T**opic model for **C**ommunity discovery, called STC, by integrating social links, content/topics, and sentiment information. The main goal of this approach is to discover sentiment-level communities, i.e., to find out some communities containing dominant sentiments on certain topics even though not all communities have dominant sentiment topics. In our STC model, we define a community as a collection of people who are directly or indirectly connected and share some sentiment topics with some members in this collection. Note that not all the topics are discussed by every member of the community, also not all the members have the identical sentiment towards a certain topic, and the connectivity among members is also a very important factor. In many cases, even if two groups of people have similar sentiment-topic distributions, they are not included in the same community when the two groups follow different user distributions. Experimental results on two types of real-world datasets demonstrate that our STC model can not only achieve comparable perfor-

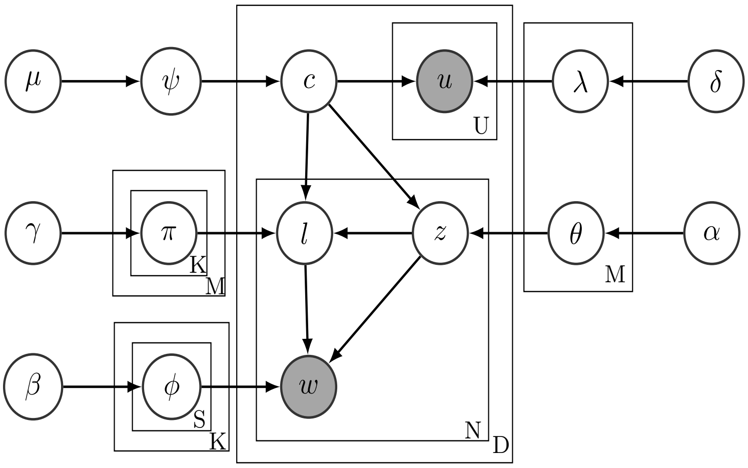


Figure 3.1: Graphical notation of our proposed STC model.

mance compared with a state-of-the-art community model, but also can identify communities with different topic-sentiment distributions.

STC model considers the users who are sharing the document as the social information, whereas the important author-recipient relationship in each document is ignored. We propose another two novel community discovery models by combining social links, author based topics and sentiment information to identify communities with different sentiment-topic distributions. One is an **A**uthor-based **S**entiment-**T**opic model for **C**ommunity discovery, called ASTC, the other is called ASTCx (the extension of the ASTC model). We evaluate our models on two real-world datasets, and the experimental results demonstrate the effectiveness of our proposed models. We also perform comparisons among STC, ASTC and ASTCx models.

Chapter outline

The reminder of this chapter is organized as follows: We introduce our first community discovery model, STC, together with the corresponding generative process and parameter estimation in Section 3.1. In Section 3.2, we report the experimental evaluation for our STC model on t-wo real-world datasets, the comparison with an up-to-date model is also reported. To overcome the weakness of STC, we propose another two novel community discovery models, ASTC and ASTCx, the details are presented in Section 3.3. In Section 3.4, we report the experimental results and performance analysis on two real-world datasets. Finally we conclude this chapter in Section 3.5.

3.1 A Social Sentiment-Topic Community Discovery Model (STC)

The graphical representation of our proposed community model, STC, is shown in Figure 3.1.

In Figure 3.1, the nodes in the circle are the random variables and the edges show the dependence between variables. Arrows in this directed graph mean the link from the parent nodes to their child nodes. The surrounding rectangular plates denote replicated sampling.

There are mainly two different kinds of variables in this model, the latent variables (shaded variables in Figure 3.1) and the observable ones (unshaded variables in Figure 3.1):

- The main latent (hidden) variables:

  Community assignment c, $c = 1, 2, \cdots, M$.

  Topic assignment z, $z = 1, 2, \cdots, K$.

  Sentiment label assignment l, $l = 1, 2, \cdots, S$.

- The main observable variables:

  Word w: the word in the document.

  Person u: the person who is sharing the document.

The reminder nodes like μ, ψ, etc. will be introduced in the following subsection.

3.1.1 Generative Process

Suppose there are K latent topics and S sentiment polarities, for each topic, and for each sentiment, we have:

$$\phi_{k,s}|\beta \sim Dir(\beta),$$

where ϕ is the topic-sentiment distribution over words.

Let M be the number of communities, each community is related to three key parameters: 1) user participant mixture λ; 2) topic mixture θ; 3) sentiment mixture π.

Specifically, in each community m ($m = 1, 2, ..., M$), θ_m is the topic mixture (proportion) for the community m, which follows a Dirichlet distribution $Dir(\alpha)$, λ_m is the user participant mixture with respect to community m, which has a Dirichlet distribution with hyperparameter δ. And $\pi_{m,k}$ is the sentiment mixture for topic k of community m. Note that the sentiments are studied based on topics, it is not reasonable to study sentiments without considering the corresponding topics. For example, given two topics “laptop” and “weather”, the sentiment words “nice” and "bad" can be used to describe both topics. It is not clear which topic is discussed by people with a sentiment word "nice" if the topic is not provided.

$$\theta_m|\alpha \sim Dir(\alpha), \quad \lambda_m|\delta \sim Dir(\delta), \quad \pi_{m,k}|\gamma \sim Dir(\gamma).$$

We define a community proportion ψ based on the whole corpus, $\psi|\mu \sim Dir(\mu)$.

In this model, α, β, δ, γ, μ are the hyperparameters of Dirichlet distributions.

Then the generative process for each document d, $d = 1, 2, ..., D$ is shown as follows:

Choose a community assignment c_d for a document d:

$$c_d|\psi \sim Mult(\psi).$$

Assume there are U_d people sharing a document d. For each person $u_{d,p}$ ($p = 1, 2, ..., U_d$) associated with document d, the generative process is:

Choose a user $u_{d,p}$ from the participant mixture of community c_d:

$$u_{d,p}|\lambda, c_d \sim Mult(\lambda_{c_d}).$$

Suppose there are N_d word tokens in a document d,

For each word token $w_{d,n}$ ($n = 1, 2, ..., N_d$) in document d. The generative process is:

1) Choose a topic assignment $z_{d,n}$ from the topic mixture of community c_d:

$$z_{d,n}|\theta, c_d \sim Mult(\theta_{c_d}).$$

2) Choose a sentiment label $l_{d,n}$ from the c_d-th community's sentiment mixture:

$$l_{d,n}|c_d, z_{d,n}, \pi \sim Mult(\pi_{c_d, z_{d,n}}).$$

3) Choose a word $w_{d,n}$ from the distribution $\phi_{k,s}$ over words defined by the topic $z_{d,n}$ and sentiment label $l_{d,n}$:

$$w_{d,n}|z_{d,n}, l_{d,n}, \phi \sim Mult(\phi_{z_{d,n}, l_{d,n}}).$$

From the graphical representation shown in Figure 3.1, the joint probability for the proposed model can be written as Eq.3.1.

$$\begin{aligned} & P(\mathbf{u}, \mathbf{c}, \mathbf{z}, \mathbf{l}, \mathbf{w}, \lambda, \psi, \theta, \pi, \phi|\delta, \mu, \alpha, \gamma, \beta) \\ & = P(\mathbf{u}|\mathbf{c}, \lambda)P(\mathbf{c}|\psi)P(\mathbf{z}|\mathbf{c}, \theta)P(\mathbf{l}|\mathbf{c}, \mathbf{z}, \pi)P(\mathbf{w}|\mathbf{z}, \mathbf{l}, \phi) \\ & \quad P(\lambda|\delta)P(\psi|\mu)P(\theta|\alpha)P(\pi|\gamma)P(\phi|\beta). \end{aligned} \tag{3.1}$$

Table 3.1: List of statistics and variables for STC model.

Statistic/Variable	**Description**
D_m	the number of documents assigned to community m;
D	the total number of documents;
$n_{m,k}$ ($n_{m,k}^{-d}$)	the number of times word tokens in the documents of community m assigned to topic k (excluding the number of word tokens assigned to topic k in document d);
$n_{m,k,s}$ ($n_{m,k,s}^{-d}$)	the number of times word tokens in the documents of community m are assigned to topic k and sentiment label s (excluding the number of word tokens assigned to topic k and sentiment label s in document d);
n_m (n_m^{-d})	the total number of word tokens in the documents of community m (excluding the number of word tokens in document d);
$n_{k,s,v}$ ($n_{k,s,v}^{-t}$)	the number of times a word v assigned to topic k and sentiment label s (excluding the word in position t);
$n_{k,s}$ ($n_{k,s}^{-t}$)	the number of times words assigned to topic k with sentiment label s (excluding the word in position t);
$f_{d,k}$	the number of word tokens in document d associated with topic k;
f_d	the total number of word tokens in document d;
$f_{d,k,s}$	the number of word tokens in document d associated with topic k and sentiment label s;
$n_{c_d,k}^{-t}$	the number of times word tokens in community c_d assigned to topic k excluding the word token in position t;
$n_{c_d,k,s}^{-t}$	the number of times word tokens in community c_d assigned to topic k and sentiment label s excluding the word in position t;
$n_{c_d}^{-t}$	the total number of word tokens in the documents of community c_d excluding the word in position t;
$g_{m,p}$ ($g_{m,p}^{-d}$)	the number of times a person p sharing the documents of community m (excluding the number of times a person p sharing the document d);
g_m (g_m^{-d})	the number of times persons sharing the documents of community m (excluding the number of persons sharing the document d);
$e_{d,p}$	the number of times a person p sharing the document d;
e_d	the number of persons who are sharing the document d;
$\mathbf{l}_{d(k)}$	the sentiment set of topic k in document d;
$\mathbf{z}_d$	the topic set of document d;
$\mathbf{u}_d$	the person set of document d.

3.1.2 Model Inference and Parameter Estimation

In STC model, a document belongs to a single community rather than multiple communities. Each document is shared by at least two people (i.e., an author and at least one recipient) to make sure there is at least one link associated with a document. Once the sender (or the author) of the document is known, the user links associated with this document will be displayed. For inference, the statistics and variables are described in Table 3.1.

Let $t = (d, n)$, the conditional posterior probability of c_d, z_t, and l_t can be written as follows.

$$\begin{aligned} &P(c_d = m | \mathbf{c}_{-d}, \mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}) \\ &\propto \frac{D_m^{-d} + \mu_m}{\sum_{j=1}^{M} \mu_j + D - 1} \times \frac{\prod_{k \in \mathbf{z}_d} \prod_{i=0}^{f_{d,k}-1} (\alpha_k + n_{m,k}^{-d} + i)}{\prod_{i=0}^{f_d - 1} (\sum_{k=1}^{K} \alpha_k + n_{m,k}^{-d} + i)} \\ &\times \prod_{k \in \mathbf{z}_d} \frac{\prod_{s \in \mathbf{l}_{d(k)}} \prod_{i=0}^{f_{d,k,s}-1} (\gamma_s + n_{m,k,s}^{-d} + i)}{\prod_{i=0}^{f_{d,k}-1} (\sum_{s=1}^{S} \gamma_s + n_{m,k,s}^{-d} + i)} \times \frac{\prod_{p \in \mathbf{u}_d} (\delta_p + g_{m,p}^{-d})}{\prod_{i=0}^{e_d - 1} (\sum_{p=1}^{P} \delta_p + g_m^{-d} + i)}. \end{aligned} \tag{3.2}$$

When the community assignment c_d for document d is obtained, for simplicity, the posterior distribution of z_t and l_t can be derived as follows.

$$\begin{aligned} &P(z_t = k, l_t = s | \mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d) \\ &\propto \frac{n_{c_d,k}^{-t} + \alpha_k}{\sum_{k=1}^{K} n_{c_d,k}^{-t} + \alpha_k} \times \frac{n_{c_d,k,s}^{-t} + \gamma_s}{\sum_{s=1}^{S} n_{c_d,k,s}^{-t} + \gamma_s} \times \frac{n_{k,s,v}^{-t} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v}^{-t} + \beta_v}. \end{aligned} \tag{3.3}$$

The detailed mathematical derivations for the above conditional posterior probabilities are shown in Appendix A.1.

The updated parameters are represented as follows:

$$\psi_m = \frac{D_m + \mu_m}{\sum_{m=1}^{M} \mu_m + D}, \quad \lambda_{m,p} = \frac{g_{m,p} + \delta_p}{\sum_{p=1}^{P} g_{m,p} + \delta_p}, \quad \theta_{m,k} = \frac{n_{m,k} + \alpha_k}{\sum_{k=1}^{K} n_{m,k} + \alpha_k},$$

$$\pi_{m,k,s} = \frac{n_{m,k,s} + \gamma_s}{\sum_{s=1}^{S} n_{m,k,s} + \gamma_s}, \quad \phi_{k,s,v} = \frac{n_{k,s,v} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v} + \beta_v}.$$

3.2 Evaluation of the STC Model

3.2.1 Evaluation Methodology

In the traditional community discovery algorithms, modularity is usually used as a metric for evaluating the performance of the detection of disjoint communities [18, 19, 71]. However, the original modularity measurement is not suitable for overlapping communities. Afterwards, several extended modularity variants were proposed for overlapping communities, such as [92, 117].

Table 3.2: Basic information for the final datasets in the experiments.

Dataset	# Docs	# Links	# Users	# Vocabulary
EnronFourUsrs	3804	38597	5623	18215
arnold-j	2441	11474	2550	14780
twitter	2247	3459	3460	7138

The above evaluation methodologies are mainly used for the community detection based on link structures. As for the community discovery by link and content, In [119], to evaluate the model performance, a modularity based community detection method is used for comparison with the proposed CUT1 and CUT2 models. In another work [81], to evaluate the detected communities, the topic probabilities and actor profiles in communities are presented, moreover the main words for the topics are reported. [111] used four metrics to measure the community discovery algorithm, 1) normalized mutual information (NMI); 2) pairwise F-measure; 3) modularity; 4) normalized cut.

In [89], the authors conduct the qualitative analysis about the distribution of topics within communities. the topic and community proportions of a user, to evaluate the detected communities, a fuzzy modularity metric is used. Furthermore, perplexity analysis is conducted. In COCOMP model [125], the authors also use perplexity as an evaluation metric. In addition, the topic distribution of communities, the main users, and topics in some communities are presented.

Our STC model is built based on both link and content, the modularity and its variants are suitable for the topology link based community discovery algorithms. Users with few ties but more common topic interests can also belong to the same community. However, according to the focus of modularity, they might be divided into different communities.

3.2.2 Experimental Setup

In the experiments, **two** types of datasets, the email dataset and the twitter microblog dataset are used. For Enron dataset[1], we randomly select five user folders, one of them called '*arnold-j*' is used for the experiment of individual user's perspective (denoted as arnold-j), and the other four folders, namely, *ermis-f*, *shively-h*, *whalley-g* and *zipper-a* are used together as a whole dataset (denoted as EnronFourUsrs). We conduct series of preprocessing work for arnold-j and Enron-

[1] http://www-2.cs.cmu.edu/~enron/

FourUsrs, like the initial duplicated email removal and the basic text mining preprocessing (stopwords removal, stemming, etc.). The second type of dataset is a twitter corpus[2], which includes 5513 tweets, covering 4 main topics, namely, Apple, Google, Microsoft, and Twitter. We kept the tweets belonging to one of the three sentiments (i.e., positive, negative and neutral), then the empty tweets and the ones without recipients are all removed. Here the user links for this Twitter dataset were mainly built based on three kinds of relations: 1) Reply, the corresponding tweet is in reply to another user's tweet, which begins with @username, each username is denoted by a unique screen name in twitter; 2) Retweet RT, in twitter, a user is allowed to re-tweet another user's tweet. A retweet starts with "RT @username", which means that it is retweeted from @username; 3) Mention, It is viewed as a Mention when the "@username" is in the body of a tweet. We also preprocess it to make the final document format the same as the EnronFourUsrs datasets. As for the four main topics in original Twitter dataset, in fact, each main topic can be divided into several subtopics. The final preprocessed datasets for our experiments are shown in Table 3.2.

As the work in [60, 61], we also use the subjectivity lexicons as prior information for model learning. Specifically, we use MPQA[3] [103] as the sentiment prior knowledge.

In our model, the initial values of the symmetric hyperparameters are set as: $\alpha = 50/K$, $\beta = \delta = \gamma = \mu = 0.1$. The collapsed Gibbs sampling algorithms are executed 500 iterations to estimate the parameters in the models. The datasets are divided into two parts, 80% of which are used for model training, and the rest are considered as held-out test set.

3.2.3 The Log-likelihood Results vs. Gibbs Sampling Iterations.

In order to observe the convergence process of the COCOMP model and our STC model, we conduct 1000 iterations on EnronFourUsrs dataset, $M = 10$, $K = 10$. As we can see from Figure 3.2 and Figure 3.3 that the log-likelihood value tends to be stable around 300-th iteration for COCOMP and 400-th iteration for STC. It indicates that the Gibbs sampling will converge around these iterations. Note that it is hard to converge under some M and K settings. For simplicity, we run Gibbs sampling 500 iterations for COCOMP and STC in the following experiments.

3.2.4 Analysis for Distributions within Communities

In our STC model, each community has multiple topics, and each topic has multiple sentiment polarities, we studied the distributions within communities on different datasets.

[2] http://www.sananalytics.com/lab/twitter-sentiment/

[3] http://www.cs.pitt.edu/mpqa/

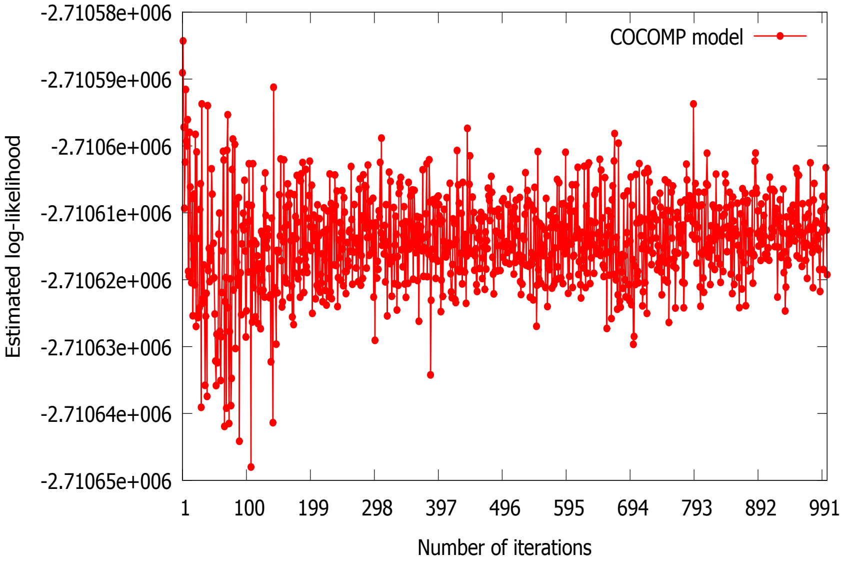


Figure 3.2: The log-likelihood results vs. Gibbs sampling iterations for COCOMP model on EnronFourUsrs dataset, $M = 10$, $K = 10$.

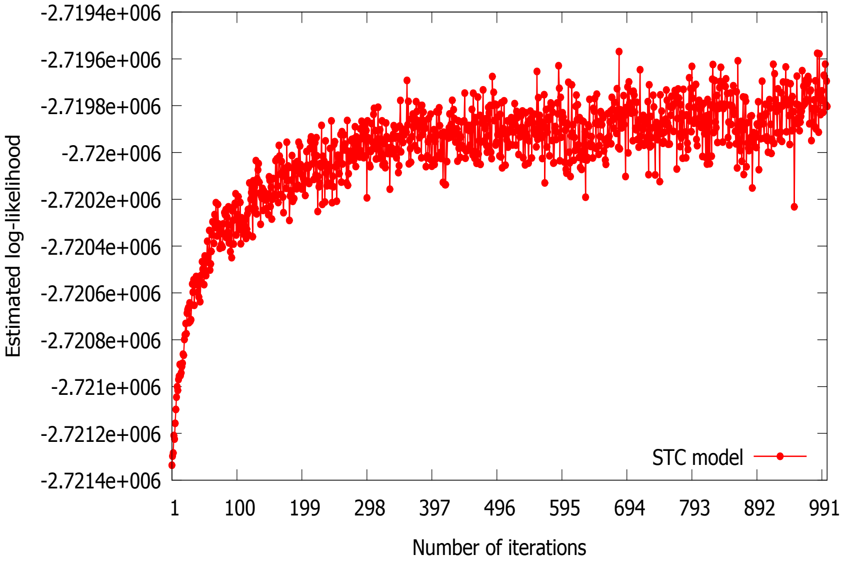
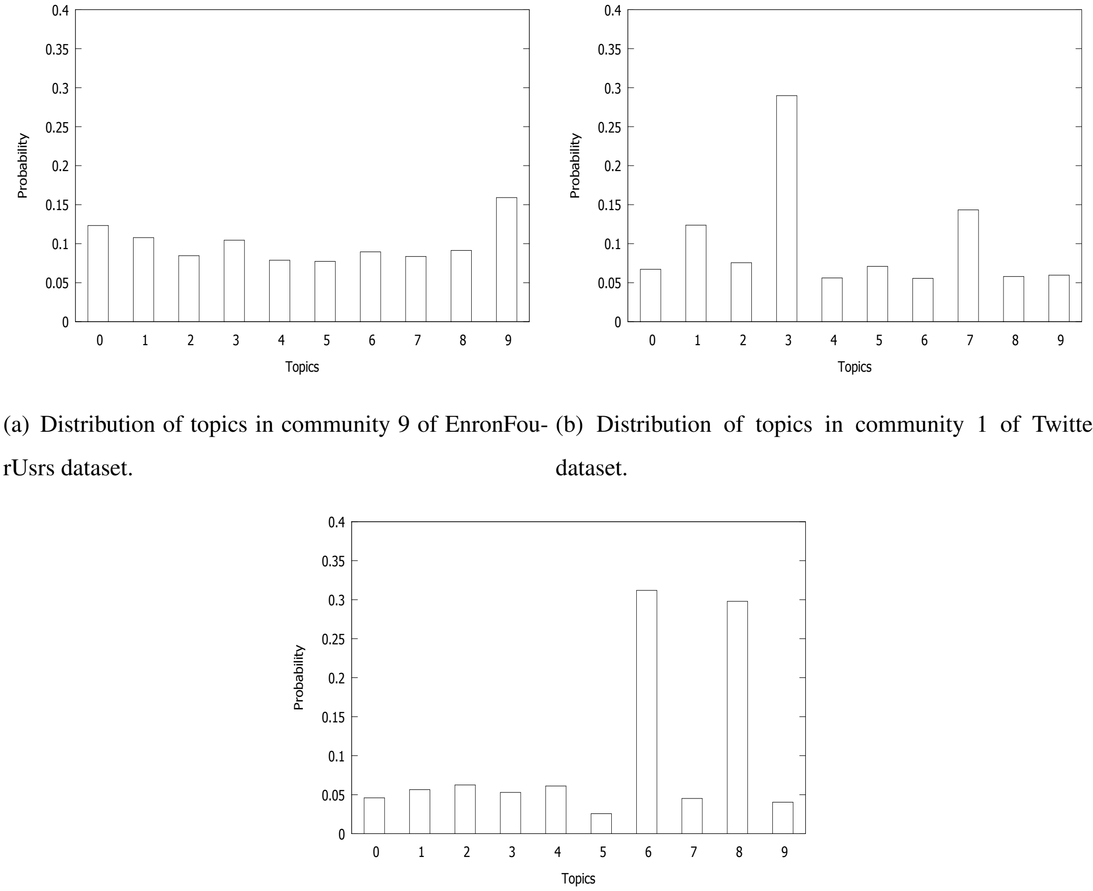


Figure 3.3: The log-likelihood results vs. Gibbs sampling iterations for STC model on EnronFourUsrs dataset, $M = 10$, $K = 10$.



(a) Distribution of topics in community 9 of EnronFourUsrs dataset.

(b) Distribution of topics in community 1 of Twitter dataset.

(c) Distribution of topics in community 13 of Twitter dataset.

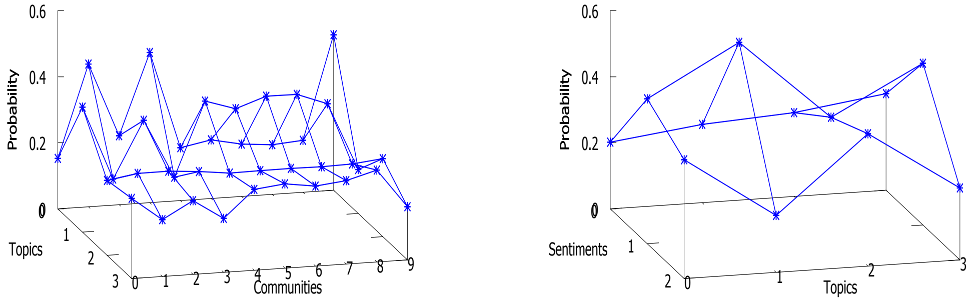
Figure 3.4: Distribution of topics in individual communities, $M = 20$, $K = 10$.

Figure 3.4 gives the distribution of topics in individual communities. It can be seen from Figure 3.4(a) that the topics are almost even within a single community 9 on EnronFourUsrs dataset. We also report selected communities on Twitter dataset, in Figure 3.4(b) and Figure 3.4(c), some topics are dominant obviously in the communities. In Figure 3.4(b), topic 3 (google android) is the dominant topic in community 1. In community 13, topic 6 (apple use) and topic 8 (iphone service) have large proportions, which are all the subtopics of "apple". These distributions imply that in some communities, people are only very interested in certain number of topics, which is in accordance with our main goal and community definition.

Apart from the analysis on the topic distribution within selected individual communities, we also investigated the topic distributions for all the communities, and the sentiment distribution for all the topics in an individual community. Figure 3.5(a) and Figure 3.5(b) give the topic and sentiment distributions on Twitter dataset, respectively. It is obvious from Figure 3.5(a) that

Table 3.3: Arnold-j's biggest community(4th community), $M = 5$, $K = 10$.

Topic ID	Topic	Positive	Negative	Neutral	people (denoted by the username of the enron email address)
4 (0.1337)	trading	0.3701	0.4498	0.1801	john.arnold (0.3746), jennifer.fraser(0.0282), ina.rangel(0.0217)
3 (0.1215)	power supply	0.5739	0.2403	0.1858	
5 (0.1167)	contract	0.3579	0.3363	0.3058	



(a) Distribution of topics in all communities for Twitter dataset.

(b) Distribution of sentiments of all topics in community 0 for Twitter dataset.

Figure 3.5: Distribution of topics within communities (sentiments for topics) for Twitter dataset, $M = 10$, $K = 4$.

different communities have nearly different topic distributions, although some topic distributions for some communities are a bit similar. As can be seen from Figure 3.5(b) about the sentiment distribution for topics in community 0 that the sentiments for different topics can be different, which is common in real-world life that two communities may have different sentiment towards certain topics even if they have similar topic distributions (i.e., the two communities are talking about similar range of topics).

Note that the community and topic number settings in Figure 3.4 and Figure 3.5 can be different. As for the Twitter dataset, it covers 4 main topics, i.e., Apple, Google, Microsoft and Twitter (mentioned in Section 3.2.2). Unlike the assumption in Figure 3.4, we suppose $M = 10$, $K = 4$ to explore the distribution of topics within communities (sentiments for topics) on Twitter dataset.

Table 3.4: Selected communities of the user @Apple (ScreenName), $M = 20$, $K = 10$.

Community	Topic ID	Topic	Positive	Negative	Neutral
9	6 (0.3075)	iphone service	0.9152	0.0492	0.0356
	8 (0.2967)	apple use	0.9398	0.0335	0.0267
10	3 (0.2895)	google android	0.8445	0.0618	0.0937
	1 (0.1327)	twitter operation	0.6029	0.1972	0.1999
5	7 (0.1373)	microsoft	0.1595	0.7182	0.1223
	2 (0.1315)	twitter share	0.6311	0.2307	0.1382

3.2.5 Community Analysis on Individual Users

We also studied the communities for a single user, arnold-j (*John Arnold*, a vice president in Enron company). Table 3.3 lists the largest community membership (community 4) for arnold-j, Column 1 and 2 show the main relevant topics and the corresponding probabilities within this community, columns 3-5 list the sentiment proportions for the corresponding topics, and the final column represents the top three active persons with high likelihoods in this community. It is obvious from Table 3.3 that the dominant sentiment polarity can vary with topics.

In Twitter dataset, we choose one entity with the screen name '@Apple' to study the hidden knowledge in its community. Table 3.4 shows the selected communities and sentiment topics that @Apple related to. Column 1 gives three selected participated communities, column 2 and 3 list the top two mainly discussed topics for each community with proportions, and the last three columns describe the sentiment proportions for the corresponding topics. It is obvious from Table 3.4 that the mainly discussed topics among communities are different, which demonstrates that community 9, 10 and 5 are well identified, and also proves the effectiveness and feasibility of our model.

Based on the topics listed in Table 3.4, we show the top five words for each sentiment polarities of *topic* 1 and *topic* 6 in Table 3.5, each column lists a collection of highly ranked sentiment words and topic words. From these words, we can observe that topic 1 is about *twitter*, and topic 6 is about *iphone service* related to *apple*. It's a first attempt to detect sentiment-topic level communities via our STC model, while the sentiment information cannot be detected by the existing COCOMP model.

Table 3.5: Top ranked words for selected topics with different sentiments extracted by STC model.

Topic 1 (Twitter Operation)			Topic 6 (Iphone Service)		
Positive	Negative	Neutral	Positive	Negative	Neutral
twitter	wrong	yeah	appl	account	touch
win	poor	custom	steve	site	babi
tech	troubl	absolut	job	close	player
world	mark	move	great	longer	feel
good	damag	launch	love	brand	report

Table 3.6: Selected communities of the user @Apple (ScreenName) by COCOMP model, M=20, K=10.

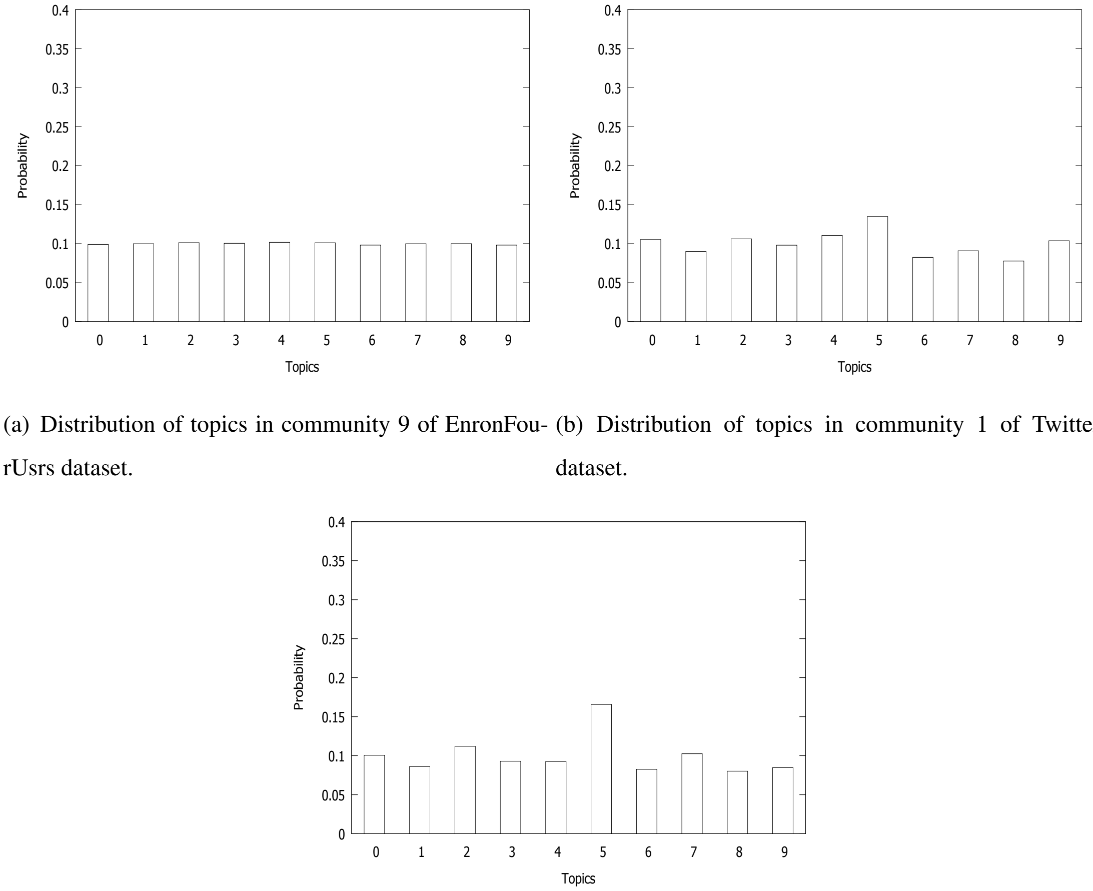
Community	Topic ID	Top topic words		
9	5(0.1503)	appl	microsoft	googl
	7(0.1104)	iphon	app	siri
10	5(0.1785)	appl	microsoft	googl
	7(0.1188)	iphon	app	siri
5	5(0.1592)	appl	microsoft	googl
	2(0.1144)	twitter	android	lol

3.2.6 Comparing with a Baseline Model: COCOMP Model

3.2.6.1 Analysis for distributions within communities by COCOMP model (baseline)

Figure 3.6 shows the distributions of topics in individual communities by COCOMP model. In Figure 3.6(a), the proportions of the 10 topics in community 9 of EnronFourUsrs dataset are very similar. In both Figure 3.6(b) and Figure 3.6(c), topic 5 is the dominant topic in the two communities on Twitter dataset.

In Figure 3.7, the distributions of topics in all communities for Twitter dataset by COCOMP model are illustrated. It is obvious that the topic distributions for the communities are very even. According to the definition of community, users with similar topic distributions tend to belong to the same communities. However, the different communities have similar topic distributions by COCOMP model.



(a) Distribution of topics in community 9 of EnronFourUsrs dataset.

(b) Distribution of topics in community 1 of Twitter dataset.

(c) Distribution of topics in community 13 of Twitter dataset.

Figure 3.6: Distribution of topics in individual communities by COCOMP model, $M = 20$, $K = 10$.

3.2.6.2 Community analysis on individual users by COCOMP model (baseline)

Table 3.6 reports the results of selected communities for the user @Apple by COCOMP model, the first column of Table 3.6 lists the selected communities, i.e., 9, 10 and 5. The second column gives the ID and proportions of the top 2 most popular topics in the corresponding communities. The last column shows the top three words for the corresponding topics. From Table 3.6, we can see that topic 5 is the most popular topic in all the three communities. Users with similar topics tend to belong to the same communities. However, communities 9, 10 and 5 have very similar topic distributions.

For the community analysis on individual users by the existing COCOMP model, the biggest community membership (4th community) for a single user, *arnold-j*, is presented in Table 3.7. The first column shows the top 3 topics discussed in this community. The top 3 topic words for

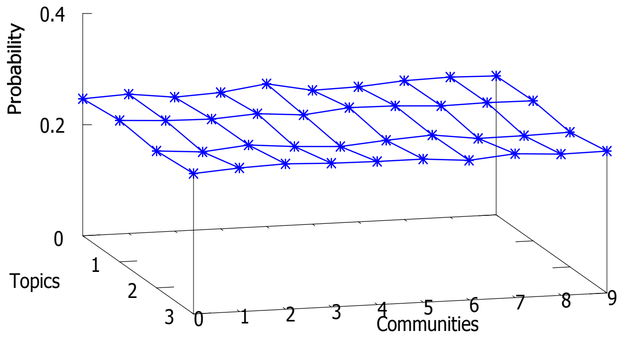


Figure 3.7: Distributions of topics in all communities for Twitter dataset by COCOMP model, $M = 10$, $K = 4$.

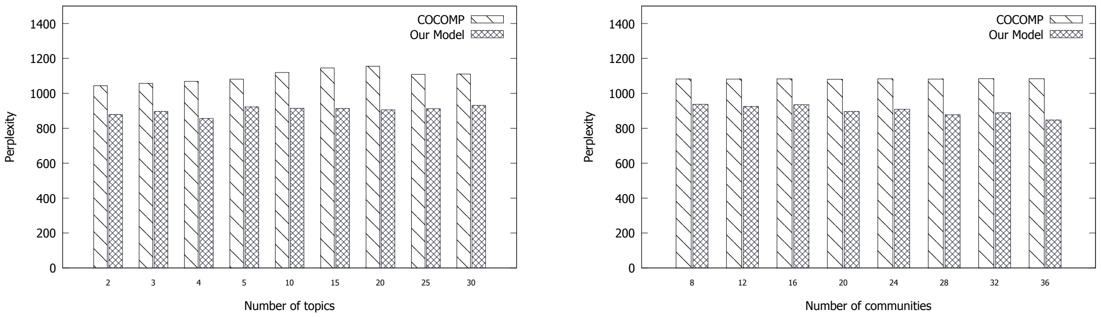
Table 3.7: Arnold-j's biggest community(4th community) by COCOMP model, M=5, K=10.

Topic ID	Top topic words	People (denoted by the username of the enron email address)
7(0.1017)	john market receiv	john.arnold (0.1146), jennifer.fraser(0.0103), dutch.quigley(0.0089)
6(0.1015)	subject arnold deal	
4(0.1014)	week work messag	

each topic are listed in the second column. The last column displays the main active users with proportions in the community. However, the sentiment information for the corresponding topics is not explored by COCOMP.

3.2.6.3 Perplexity value comparison: STC vs. COCOMP model

Note that the ground-truth communities are usually unavailable, which make the evaluation challenging. To evaluate our model, we also analysed the perplexity value, and made comparison with the state-of-the-art COCOMP model [125], which is a topic-level community discovery model. Each word in our model is determined by two factors, namely topic and sentiment, while there is only one factor, topic, for the COCOMP model. In our STC model, to generate a target word, both the topic and sentiment should be correctly assigned, otherwise the perplexity value will get worse, while only a correct topic assignment is required in COCOMP model. The computation equation for the perplexity of our STC model is shown in Eq.3.4. The lower perplexity tends to



(a) Perplexity under varying number of topics, $M = 20$.

(b) Perplexity under varying number of communities, $K = 5$.

Figure 3.8: Perplexity results comparison between COCOMP and our STC model for Twitter dataset.

have the better performance.

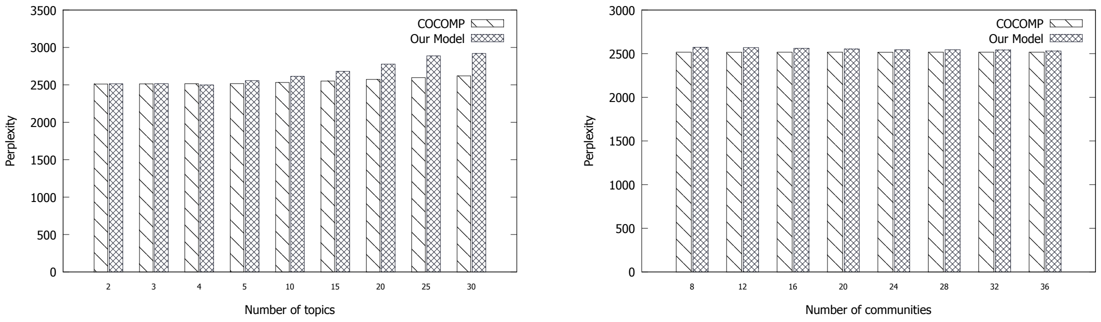
$$Perplexity_{\_STC}(D_{test}) = \exp\left\{-\frac{\sum_{m=1}^{M}\log Pro(\tilde{\mathbf{w}}_m|\mathbf{w})}{\sum_{m=1}^{M} n_m}\right\}. \tag{3.4}$$

$$\begin{aligned} & Pro(\tilde{\mathbf{w}}_m|\mathbf{w}) \\ & = \prod_{n=1}^{n_m}\sum_{k=1}^{K}\sum_{s=1}^{S} Pro(w_n = t|z_n = k, l_n = s) \times Pro(l_n = s|z_n = k, c_{w_n} = m) \times Pro(z_n = k|c_{w_n} = m) \\ & = \prod_{t=1}^{V}\left(\sum_{k=1}^{K}\sum_{s=1}^{S}\phi_{k,s,t}\pi_{m,k,s}\theta_{m,k}\right)^{n_m^{(t)}}. \end{aligned} \tag{3.5}$$

$$\log Pro(\tilde{\mathbf{w}}_m|\mathbf{w}) = \sum_{t=1}^{V} n_m^{(t)} \log(\sum_{k=1}^{K}\sum_{s=1}^{S}\phi_{k,s,t}\pi_{m,k,s}\theta_{m,k}). \tag{3.6}$$

In Eq.3.4, D_{test} shows the held-out testing documents, $\tilde{\mathbf{w}}_m$ denotes the words from testing documents appeared in community m, $\mathbf{w}$ represents the words in the training documents. n_m is the number of words in community m. As for Eq.3.5, $n_m^{(t)}$ is the number of times a term t observed in community m, and c_{w_n} represents the community that the word w_n appears in.

The perplexity results for the two datasets are shown in Figure 3.8 and Figure 3.9. In each figure we illustrated the values of perplexity for our STC model and COCOMP with varying number of topics and communities. As can be seen from Figure 3.8(a) and Figure 3.8(b), the perplexity values of our model are lower than the COCOMP model. Statistical significance difference is at 5% significance level by using a two-tailed paired t-test. Although in Figure 3.9(a) and Figure 3.9(b), the perplexity values for STC are a little higher than the COCOMP, it is still overall



(a) Perplexity under varying number of topics, $M = 20$.

(b) Perplexity under varying number of communities, $K = 5$.

Figure 3.9: Perplexity results comparison between COCOMP and our STC model for EnronFourUsrs dataset.

comparable to the COCOMP model. Enron email and Twitter are two different types of social networking sites, the former is more formal than the latter. Generally, there are more sentiment information in tweets than in emails. It is not the main concerning about which model has better perplexity value as long as our model has closer performance with COCOMP. Our model is proposed to identify sentiment level communities, which is not considered by COCOMP and other community discovery methods.

3.2.7 Discussions

We build our community discovery model, STC, by using social links, topics and sentiment information in a unified way. Those three factors are very significant to the identification of the meaningful community structures. However, it is not indicating that the more additional information incorporated into the model, the better result we can get. When the information is not important, the redundant factors can make the model more complex and inefficient. Not all the communities have sentiment information, our model is proposed to identify communities that have a certain degree of sentiment polarities.

3.3 Joint Social Link and Author-Based Sentiment-Topic Community Discovery Models

3.3.1 Preliminaries

In this section, to make full use of the sentiment factor for finding more meaningful and valuable community structures, we present the new definition and assumptions as follows.

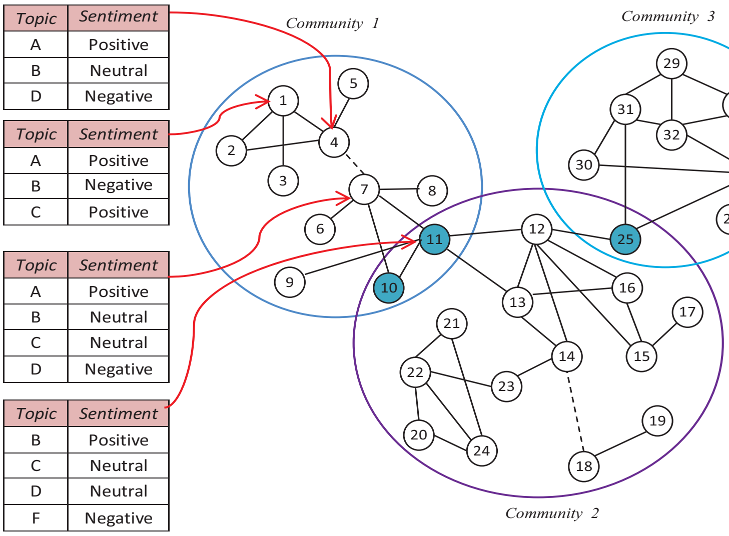


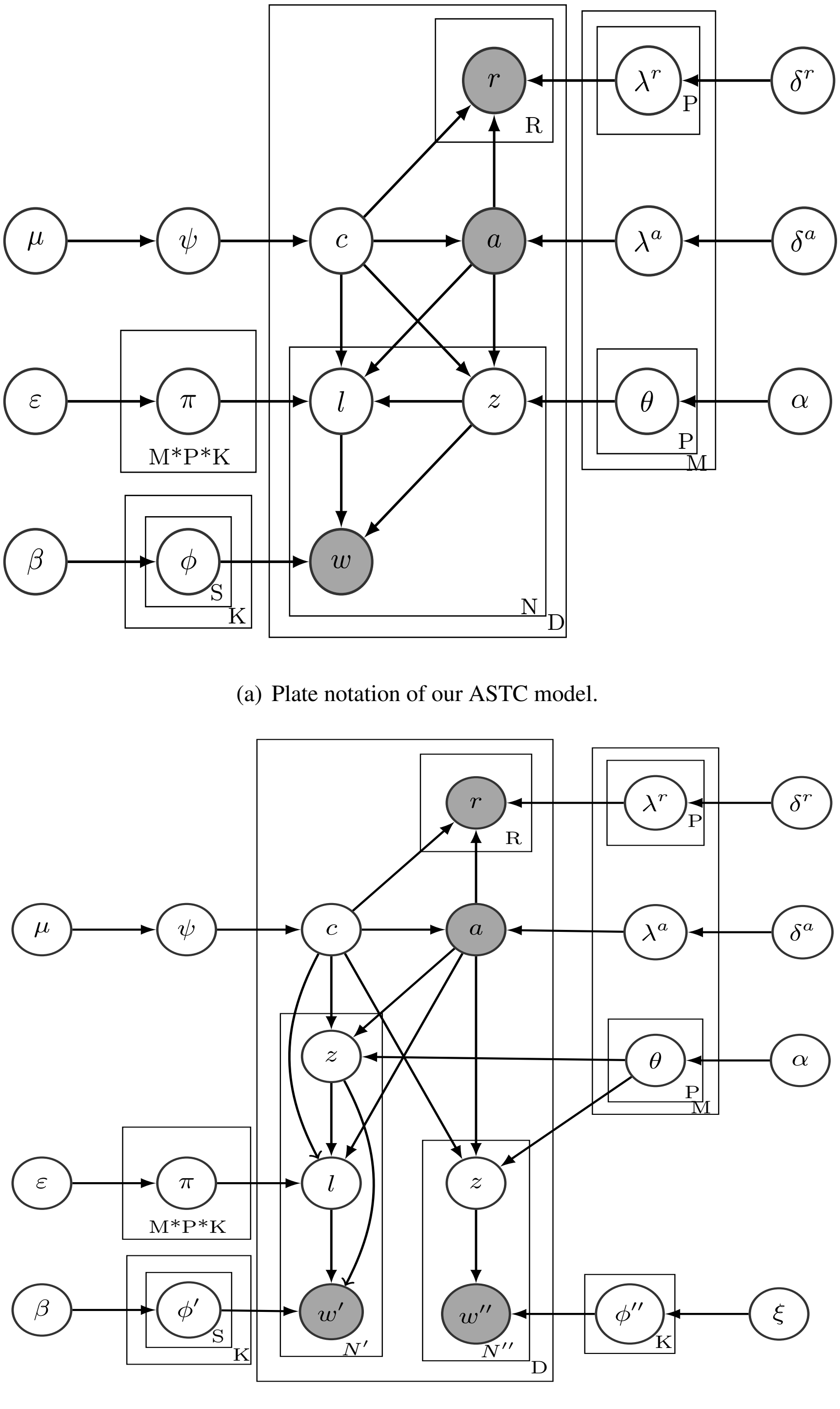
Figure 3.10: An example of communities identified from a network.

Definition 1 (Social Community) *A social community is composed of a group of people, who are connected or indirectly connected, and share some sentiment topics with some members of this group.*

In fact, some communities have sub-groups. And the sub-groups in a community may share some sentiment-topics even though there is no real connection between them, which is very common in Twitter. Actually, we can consider there exist virtual links between those sub-groups. Figure 3.10 illustrates an example for a network with communities. In Figure 3.10, there are three communities with overlapping nodes (users), namely, 10, 11 and 25. In community 1, the dotted line between node 4 and node 7 means it is a virtual connection.

Assumption 1 (Community's Distributions) *Generally, the topics of a message are determined by senders (or authors). Each community is associated with four distributions, namely, author distribution, author's recipient distribution, author's topic distribution, and author's sentiment distribution towards a certain topic.*

Communities differ from each other in terms of different community distributions. In some communities, two sub-groups follow the same distribution, although there is no real link between them. User 4 and 7 in Figure 3.10 follow the identical community distributions, and they have common topics A, B and D with similar sentiment. So we can consider that there exists a virtual connection between them. If we use traditional methods, user 4 and user 7 are likely to be separated into two different communities. We can observe that user 10 and user 11 belong to both community 1 and



(a) Plate notation of our ASTC model.

(b) Plate notation of our ASTCx model.

Figure 3.11: Plate notation of our proposed models.

2 simultaneously, which means they have more social circles than other users. Although user 1 and user 4 are connected each other in the same community and sharing two topics A and B, their sentiment towards the same topic is not necessarily the same, for topic B, the sentiment of user 1 is negative, while it is neutral for user 4, which is normal in real world.

Assumption 2 (T-S Unambiguous Community) *It is not uncommon that there is no clear sentiment polarity on topics in some communities. Even two densely connected people can have disagreement with respect to a certain topic in a community. We are more interested in the communities containing clear sentiment inclines towards corresponding topics, which are considered as Topic-Sentiment (T-S) unambiguous communities.*

3.3.2 Our New Community Discovery Models

We present our two new community models, ASTC and ASTCx in the form of plate notations in Figure 3.11. In the two sub-figures, the shaded nodes are observed, while the unshaded ones are hidden. The variables, hyperparameters, and parameters are shown as follows.

Main Variables 1) Word w (w' or w''): the word in the document (the sentiment word or the topic word). 2) Author a: the person who is the creator or the sender of the document. 3) Recipient r: the person who is sharing the document except the author. 4) c: community assignment. 5) z: topic assignment. 6) l: sentiment assignment.

In the two models, there exists a link from z to l, which means sentiments are associated with corresponding topics.

Hyperparameters α, β, δ^a, δ^r, ε, μ and ξ are used as hyperparameters of Dirichlet distributions.

Parameters We assume there are M communities, K latent topics and S sentiment polarities. $Dir(\cdot)$ represents Dirichlet distribution and $Mult(\cdot)$ denotes multinomial distribution.

- ϕ and ϕ': the topic-sentiment distribution over words, $\phi_{k,s}|\beta \sim Dir(\beta)$, and $\phi'_{k,s}|\beta \sim Dir(\beta)$; ϕ'': the topic distribution over words, $\phi''_k|\xi \sim Dir(\xi)$.
- $\theta_{m,p}$: the topic mixture (proportion) for the author p in community m, $\theta_{m,p}|\alpha \sim Dir(\alpha)$.
- λ^a_m: the author participant mixture in community m, which has a Dirichlet distribution with $\lambda^a_m|\delta^a \sim Dir(\delta^a)$.

- $\lambda_{m,p}^r$: the recipient participant mixture with respect to the author p in community m, $\lambda_{m,p}^r|\delta^r \sim Dir(\delta^r)$.

- $\pi_{m,p,k}$: the sentiment mixture with respect to topic k of the author p in community m, $\pi_{m,p,k}|\varepsilon \sim Dir(\varepsilon)$.

- ψ: the community proportion based on the whole corpus, $\psi|\mu \sim Dir(\mu)$.

3.3.3 Generative Process

For the networked data, a document contains two parts: 1) the users who are sharing the documents (including the author). 2) the text of the document. For our ASTCx model, we present the generative process for users and text of each social document d ($d = 1, 2, ..., D$) in Figure 3.12 and Figure 3.13, respectively.

The generative process for the ASTC model is similar to that of ASTCx model, we will not describe it in detail.

Draw a community assignment c_d for a document d: $c_d|\psi \sim Mult(\psi)$.
Draw an author a_d from the author distribution of community c_d : $a_d|\lambda^a, c_d \sim Mult(\lambda_{c_d}^a)$.
Draw each recipient $r_{d,p}$ ($p = 1, 2, ..., R_d$) of document d from the recipient mixture of author a_d in community c_d: $r_{d,p}|\lambda^r, c_d, a_d \sim Mult(\lambda_{c_d,a_d}^r)$.

Figure 3.12: The generative process for the users of a document.

3.3.4 Inference and Parameter Estimation

$$
\begin{aligned}
A = & \frac{D_m^{-d} + \mu_m}{\sum_{j=1}^{M} \mu_j + D - 1} \cdot \frac{\delta_p^a + g_{m,p}^{-d}}{\sum_{p=1}^{P_a} \delta_p^a + g_m^{-d}} \\
& \cdot \frac{\prod_{p' \in \mathbf{r}_d} (\delta_{p'}^r + g_{m,a_d,p'}^{-d})}{\prod_{i=0}^{R_d-1} (\sum_{p'=1}^{P_r} \delta_{p'}^r + g_{m,a_d}^{-d} + i)} \\
& \cdot \frac{\prod_{k \in \mathbf{z}_d} \prod_{i=0}^{f_{d,k}-1} (\alpha_k + n_{m,a_d,k}^{-d} + i)}{\prod_{i=0}^{f_d-1} (\sum_{k=1}^{K} \alpha_k + n_{m,a_d,k}^{-d} + i)}
\end{aligned}
\tag{3.7}
$$

Table 3.8 lists the main statistics and variables for the model inference, where we use $t = (d, n)$ to denote a word token at the n-th position in document d. In our ASTCx model, the conditional posterior probability of c_d is shown in Eq. 3.8; If the word t is a sentiment word, the inference for z_t and l_t can be written as Eq. 3.9.

Table 3.8: Notations for statistics and variables for the two new models.

Symbol	**Description** // comm.(community), doc.(document)
D_m (D_m^{-d})	the number of docs. assigned to comm. m (excluding doc. d);
D	the total number of docs.;
$n_{m,p,k}$ ($n_{m,p,k}^{-d}$)	the number of times word tokens in the docs. of author p from comm. m are assigned to topic k (excluding the number of times word tokens assigned to topic k in the doc. d); //$n_{m,a_d,k}^{-d}$, when p=a_d
$n'^{-d}_{m,p,k}$	the number of times sentiment word tokens in the docs. of author p from comm. m are assigned to topic k excluding the number of times sentiment word tokens assigned to topic k in the doc. d; //$n'^{-d}_{m,a_d,k}$, when p=a_d
$n'_{m,p,k,s}$ ($n'^{-d}_{m,p,k,s}$)	the number of times sentiment word tokens in the docs. of author p from comm. m are assigned to topic k and sentiment label s (excluding the number of times sentiment word tokens assigned to topic k and sentiment label s in the doc. d); //$n'^{-d}_{m,a_d,k,s}$, when p=a_d
$n_{m,p}$ ($n^{-d}_{m,p}$)	the total number of words in the docs. of author p from comm. m (excluding those in doc. d); //n^{-d}_{m,a_d}, when p=a_d
$n_{k,s,v'}$ ($n^{-t}_{k,s,v'}$)	the number of times a sentiment word v' is assigned to topic k and sentiment label s (excluding the word in position t);
$n_{k,v''}$ ($n^{-t}_{k,v''}$)	the number of times a non-sentiment word v'' is assigned to topic k (excluding the word in position t);
$n_{k,s}$ ($n^{-t}_{k,s}$)	the number of times words are assigned to topic k with sentiment label s (excluding the word in position t);
$f_{d,k}$	the number of word tokens in doc. d associated with topic k;
$f'_{d,k}$	the number of sentiment word tokens in doc. d associated with topic k;
f_d	the total number of words in doc. d;
$f'_{d,k,s}$	the number of sentiment word tokens in doc. d associated with topic k and sentiment label s;
$n^{-t}_{c_d,a_d,k}$	the number of times word tokens of author a_d from comm. c_d are assigned to topic k excluding the word in position t;
$n'^{-t}_{c_d,a_d,k,s}$	the number of times sentiment word tokens of author a_d from comm. c_d are assigned to topic k and sentiment label s excluding the word in position t;
$n^{-t}_{c_d,a_d}$	the total number of words in the docs. of author a_d from comm. c_d excluding the word in position t;
$g_{m,p}$ ($g^{-d}_{m,p}$)	the number of times an author p sharing the docs. of comm. m (excluding the number of times an author p sharing the doc. d); //g^{-d}_{m,a_d}, when p=a_d
$g_{m,p,p'}$ ($g^{-d}_{m,p,p'}$)	the number of times a user p' sharing the docs. of author p from comm. m (excluding the number of times a user p' sharing the doc. d); //$g^{-d}_{m,a_d,p'}$, when p=a_d
g_m (g_m^{-d})	the number of times persons sharing the docs. of comm. m (excluding the number of times persons sharing the doc. d);
R_d	the number of recipients who are sharing the doc. d;
$\mathbf{l}'_{d_{(k)}}$	the sentiment set of topic k in doc. d;
$\mathbf{z}_d$	the topic set of doc. d;
$\mathbf{z}'_d$	the topic set for sentiment words of doc. d;
$\mathbf{r}_d$	the recipient user set of doc. d.

Assume that a document d contains N_d word tokens including N'_d sentiment words and N''_d topic words.

(i) The generative process for each sentiment word token $w'_{d,n'}$ ($n' = 1, 2, ..., N'_d$) in document d:

- Based on topic mixture θ, sample a topic assignment $z_{d,n'}$ of author a_d in community c_d: $z_{d,n'}|\theta, c_d, a_d \sim Mult(\theta_{c_d,a_d})$.
- Draw a sentiment assignment $l_{d,n'}$ from the a_d-th author's sentiment mixture in community c_d: $l_{d,n'}|c_d, a_d, z_{d,n'}, \pi \sim Mult(\pi_{c_d,a_d,z_{d,n'}})$.
- Based on the topic assignment $z_{d,n'}$ and sentiment $l_{d,n'}$, sample a sentiment word $w'_{d,n'}$ following the distribution $\phi'_{k,s}$: $w'_{d,n'}|z_{d,n'}, l_{d,n'}, \phi' \sim Mult(\phi'_{z_{d,n'},l_{d,n'}})$.

(ii) The generative process for each topic word token $w''_{d,n''}$ ($n'' = 1, 2, ..., N''_d$) in document d:

- Choose a topic assignment $z_{d,n''}$ using the same way as that for $z_{d,n'}$.
- According to the topic assignment $z_{d,n''}$, choose a topic word $w''_{d,n''}$ from the distribution ϕ''_k over words: $w''_{d,n''}|z_{d,n''}, \phi'' \sim Mult(\phi''_{z_{d,n''}})$.

Figure 3.13: The generative process for the text (words) of a document.

$$
\begin{aligned}
& P(c_d = m|\mathbf{c}_{-d}, \mathbf{r}, \mathbf{a}, \mathbf{z}, \mathbf{l}, \mathbf{w}', \mathbf{w}'') \\
& \propto A \cdot \prod_{k \in \mathbf{z}'_d} \frac{\prod_{s \in \mathbf{l}'_{d_{(k)}}} \prod_{i=0}^{f'_{d,k,s}-1} (\varepsilon_s + n'^{-d}_{m,a_d,k,s} + i)}{\prod_{i=0}^{f'_{d,k}-1} (\sum_{s=1}^{S} \varepsilon_s + n'^{-d}_{m,a_d,k,s} + i)}
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
& P(z_t = k, l_t = s|\mathbf{w}', \mathbf{w}'', \mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d, a_d) \\
& \propto \frac{n^{-t}_{c_d,a_d,k} + \alpha_k}{\sum_{k=1}^{K} n^{-t}_{c_d,a_d,k} + \alpha_k} \cdot \frac{n'^{-t}_{c_d,a_d,k,s} + \varepsilon_s}{\sum_{s=1}^{S} n'^{-t}_{c_d,a_d,k,s} + \varepsilon_s} \\
& \cdot \frac{n^{-t}_{k,s,v'} + \beta_{v'}}{\sum_{v'=1}^{V'} n^{-t}_{k,s,v'} + \beta_{v'}}
\end{aligned}
\tag{3.9}
$$

When it comes to a topic word t, the conditional posterior probability of z_t can be denoted as Eq. 3.10.

$$
\begin{aligned}
& P(z_t = k|\mathbf{w}', \mathbf{w}'', \mathbf{z}_{-t}, c_d, a_d) \\
& \propto \frac{n^{-t}_{c_d,a_d,k} + \alpha_k}{\sum_{k=1}^{K} n^{-t}_{c_d,a_d,k} + \alpha_k} \cdot \frac{n^{-t}_{k,v''} + \xi_{v''}}{\sum_{v''=1}^{V''} n^{-t}_{k,v''} + \xi_{v''}}
\end{aligned}
\tag{3.10}
$$

For the ASTC model, once the community assignment c_d for a document d is worked out by Eq.

3.11, we can get the posterior distribution of z_t and l_t according to Eq. 3.12.

$$\begin{aligned} & P(c_d = m|\mathbf{c}_{-d}, \mathbf{r}, \mathbf{a}, \mathbf{z}, \mathbf{l}, \mathbf{w}) \\ & \propto A \cdot \prod_{k \in \mathbf{z}_d} \frac{\prod_{s \in \mathbf{l}_{d_{(k)}}} \prod_{i=0}^{f_{d,k,s}-1} (\varepsilon_s + n_{m,a_d,k,s}^{-d} + i)}{\prod_{i=0}^{f_{d,k}-1} (\sum_{s=1}^{S} \varepsilon_s + n_{m,a_d,k,s}^{-d} + i)} \end{aligned} \tag{3.11}$$

$$\begin{aligned} & P(z_t = k, l_t = s|\mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d, a_d) \\ & \propto \frac{n_{c_d,a_d,k}^{-t} + \alpha_k}{\sum_{k=1}^{K} n_{c_d,a_d,k}^{-t} + \alpha_k} \cdot \frac{n_{c_d,a_d,k,s}^{-t} + \varepsilon_s}{\sum_{s=1}^{S} n_{c_d,a_d,k,s}^{-t} + \varepsilon_s} \\ & \cdot \frac{n_{k,s,v}^{-t} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v}^{-t} + \beta_v} \end{aligned} \tag{3.12}$$

For parameter estimation, in ASTCx model the parameters, ψ_m, $\theta_{m,p,k}$, $\pi_{m,p,k,s}$, $\phi'_{k,s,v'}$, $\phi''_{k,v''}$, $\lambda^a_{m,p}$, and $\lambda^r_{m,p,p'}$ can be updated by the following expressions.

$$\begin{aligned} \psi_m &= \frac{D_m + \mu_m}{\sum_{m=1}^{M} \mu_m + D}, \quad \theta_{m,p,k} = \frac{n_{m,p,k} + \alpha_k}{\sum_{k=1}^{K} n_{m,p,k} + \alpha_k}, \\ \pi_{m,p,k,s} &= \frac{n'_{m,p,k,s} + \varepsilon_s}{\sum_{s=1}^{S} n'_{m,p,k,s} + \varepsilon_s}, \quad \phi'_{k,s,v'} = \frac{n_{k,s,v'} + \beta_{v'}}{\sum_{v'=1}^{V'} n_{k,s,v'} + \beta_{v'}}, \\ \phi''_{k,v''} &= \frac{n_{k,v''} + \xi_{v''}}{\sum_{v''=1}^{V''} n_{k,v''} + \xi_{v''}}, \quad \lambda^a_{m,p} = \frac{g_{m,p} + \delta^a_p}{\sum_{p=1}^{P_a} g_{m,p} + \delta^a_p}, \\ \lambda^r_{m,p,p'} &= \frac{g_{m,p,p'} + \delta^r_{p'}}{\sum_{p'=1}^{P_r} g_{m,p,p'} + \delta^r_{p'}}. \end{aligned}$$

Most parameters of ASTC model are similar to ASTCx, we will not report the update equation for ASTC model.

3.4 Experimental Results and Analysis

3.4.1 Experimental Settings

Two datasets are used to evaluate our models, and the detailed description is shown as follows.

Enron dataset. The Enron email dataset[4] contains a large collective of email messages. In this task, we use a short version[5] of this corpus, which is composed of 1702 messages. We removed the non-sentiment emails and did basic text preprocessings, then the final dataset used in the experiment consists of 301 documents belonging to 93 authors. There are 1281 social links associated with send-receive interactions, and the total number of recipient is 593.

[4] http://www-2.cs.cmu.edu/˜enron/

[5] http://www.sims.berkeley.edu/˜atf/enron_with_categories.tar.gz

Twitter-x dataset. The Twitter-x dataset used in the experiment is Sanders-Twitter Sentiment Corpus[6], which includes 5513 tweets, covering 4 main topics, namely, Apple, Google, Microsoft, and Twitter. We kept the tweets belonging to one of the three sentiments (i.e., positive, negative and neutral), then the tweets without recipients are removed. Further the same text preprocessing was conducted as Enron dataset. Finally, we got 1635 tweets for our experiment, and there are 2617 links from 1482 authors and 1110 recipients.

Note that this Enron dataset contains more sentiment documents than that used in STC model evaluation, so we use this dataset instead. The Twitter-x dataset used here is similar to that used on STC model. The difference is that the stemming procedure is conducted in text preprocessing for the experiments on STC model in Section 3.2. Because some words can be displayed in an incomplete style after stemming procedure, so we make text preprocessing without stemming procedure for the evaluation on ASTC and ASTCx models.

For parameter estimation, the collapsed Gibbs sampling algorithms are executed 1000 iterations. In model learning, we use MPQA[7] [103], a subjectivity lexicon, as the sentiment prior knowledge, which is also used in [60, 61]. We set the initial values of the symmetric hyperparameters as $\alpha = 50/K$, and the equal value 0.1 for β, δ^a,δ^r, ε, μ, and ξ. We set the topic number K as $4 \leq K \leq 10$, and the community number M as $10 \leq M \leq 20$ according to the datasets used in this task.

In ASTCx model, we use WordNet[8] to separate the adjectives and adverbs from each document before the model construction. In addition to the adjectives and adverbs, some verbs also can convey sentiment information, such as the words *love*, *praise*. Due to the small scale of such verbs, we won't consider them in this task.

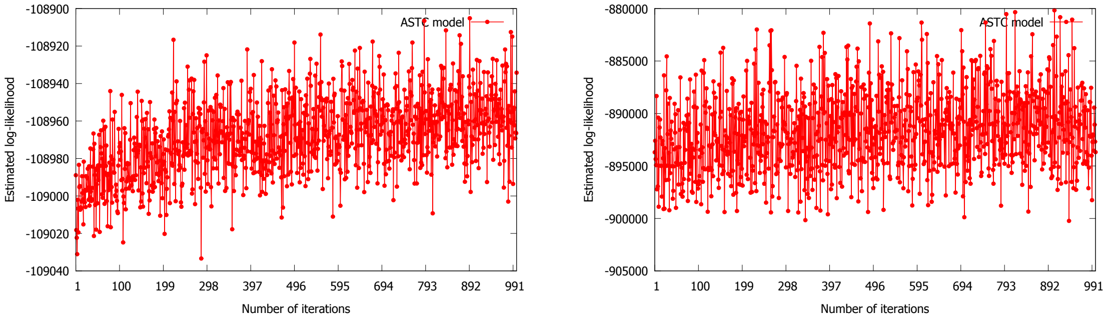
3.4.2 The Log-likelihood Results vs. Gibbs Sampling Iterations for ASTC and ASTCx Models.

We also conducted convergence analysis on Gibbs sampling process for our ASTC and ASTCx models. Figure 3.14(a) shows the log-likelihood results versus Gibbs sampling iterations on Twitter-x dataset, $M = 10$, $K = 4$. We can see that the log-likelihood value is getting stable around 450-th iteration for ASTC model. It means the Gibbs sampling will converge around this iteration. Figure 3.14(b) illustrates the log-likelihood results versus Gibbs sampling iterations for

[6] http://www.sananalytics.com/lab/twitter-sentiment/

[7] http://www.cs.pitt.edu/mpqa/

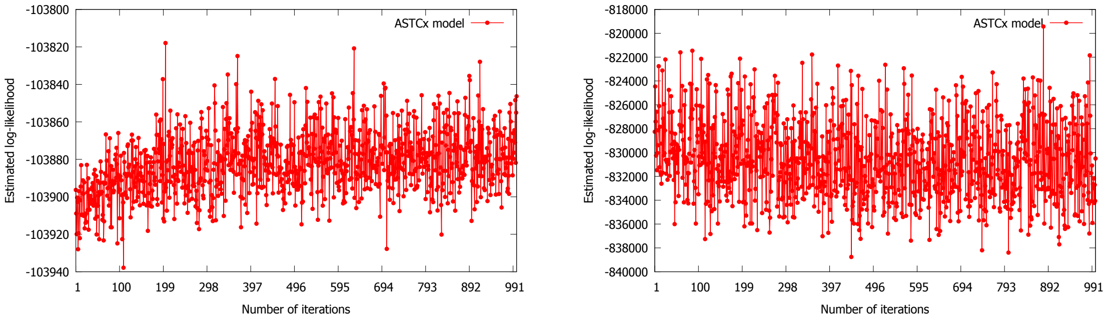
[8] http://wordnet.princeton.edu/



(a) The log-likelihood results on Twitter-x dataset, $M = 10$, $K = 4$.

(b) The log-likelihood results on Enron dataset, $M = 10$, $K = 20$.

Figure 3.14: The log-likelihood results vs. Gibbs sampling iterations for ASTC model.



(a) The log-likelihood results on Twitter-x dataset, $M = 10$, $K = 4$.

(b) The log-likelihood results on Enron dataset, $M = 10$, $K = 20$.

Figure 3.15: The log-likelihood results vs. Gibbs sampling iterations for ASTCx model.

ASTC on Enron dataset, $M = 10$, $K = 20$. It can be observed that the sampling will converge around 350-th iteration.

As for our ASTCx model, It is clear from both Figure 3.15(a) and Figure 3.15(b) that the Gibbs sampling will converge around 300-th iteration on Twitter-x and Enron datasets.

Note that it is hard to converge under some unsuitable M and K settings. For simplicity, we run Gibbs sampling 1000 iterations for ASTC, ASTCx and baseline models in our experiments.

3.4.3 Community Analysis

We conducted series of performance analysis for the detected communities and studied the effectiveness of our proposed models in this subsection. For brevity, we mainly report the performance of the ASTCx model.

3.4.3.1 Topic distribution in individual communities

There are usually multiple topics discussed in each community. While the topic proportions in each community are generally different. Figure 3.16 shows the topic distributions in some of author 81's participated communities on the Enron dataset. We represent their distributions with probability. As can be seen from Figure 3.16 that the topic distributions in the three communities are apparently different. In Figure 3.16(a), topic 9 is about marketing & trading, which is mainly discussed while other topics are either rarely involved or not really discussed with very low probabilities. Likewise, topic 0 (office work) is dominant in community 7. In Figure 3.16(c), there are more major topics involved in community 8.

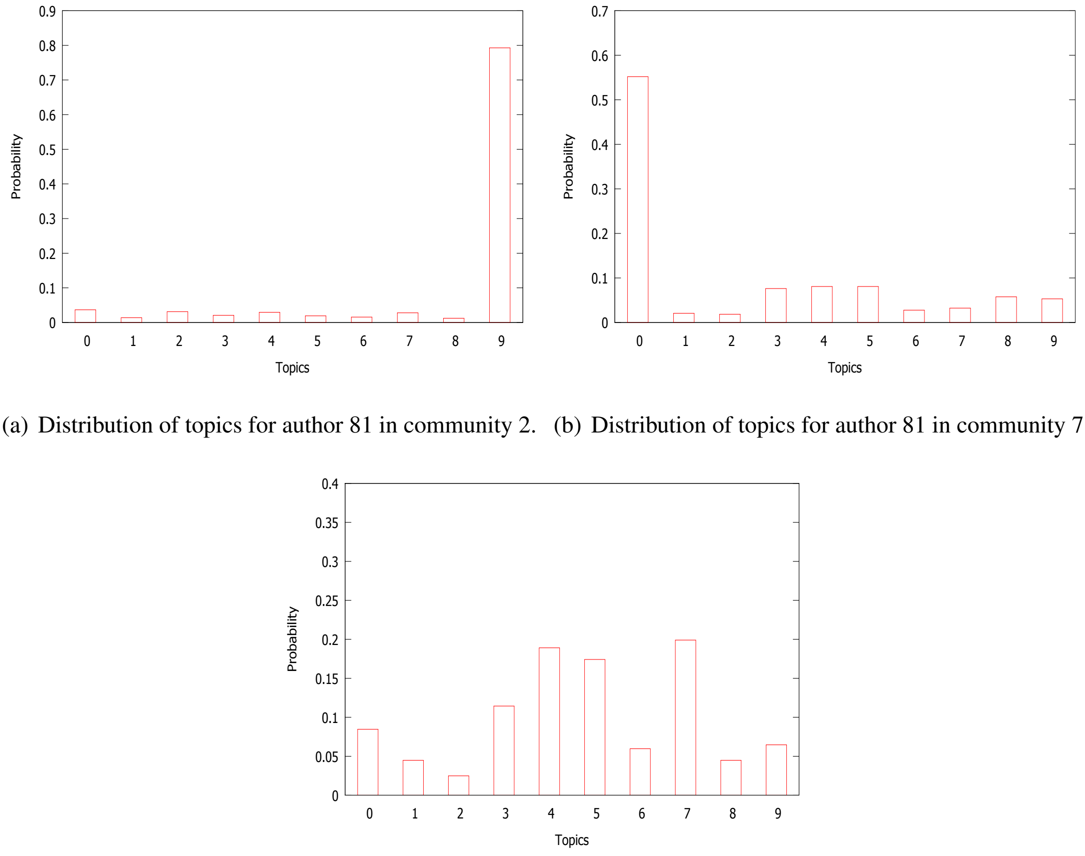
It is in accordance with the intuition that the topic distributions in each community are usually different, and our ASTCx model (likewise ASTC model) can discover communities including different topic distributions for the authors.

In real life, the topic and topic distributions for some users in individual communities are more valuable. However, as for COCOMP and STC, the topic distribution of a certain author is not studied.

3.4.3.2 Active authors' topic proportions in the same community

In our two models, each topic is drawn from an author's topic distribution in a community. Also each author's major interested topics can be different in a community.

For each dataset used in our experiment, we present some active authors' topic distributions within the same community in Figure 3.17. In Figure 3.17(a), author 1292 and author 375 in



(a) Distribution of topics for author 81 in community 2. (b) Distribution of topics for author 81 in community 7.

(c) Distribution of topics for author 81 in community 8.

Figure 3.16: Topic distribution of author-81 in some of the participated communities on Enron dataset, $M = 10$, $K = 10$.

community 10 have many common topics (T2,T3,T4 and T10) with similar proportions, although they have their respective dominant topics, T5 (technology) and T6 (iphone), respectively. And in Figure 3.17(b), *Vince Kaminski* and *Jeff Dasovich* have the same dominant topic T0, and other topics are rarely discussed or nearly absent. The author *Steven Kean* involves in all topics with nearly even distributions. In fact, *Steven Kean* is expected to related to many topics for the reason that he is a vice president and chief staff (refer to the Enron Employee Status[9]) in the Enron corporation taking charging of many things. Overall, the four authors in Figure 3.17(b) have similar topic proportions on topic T8 and T9.

Vince Kaminski is a risk management head, the main topics are T0 (office work) and T5 (employment). *Jeff Dasovich* is a government relation executive, T0 is also the mainly related topic. We don't know *John Shelk*'s position in Enron corporation, we can see from 3.17(b) that his dom-

[9] http://isi.edu/~adibi/Enron/Enron_Employee_Status.xls

inant topic is T7 (meeting).

Generally, the active users are the main people playing important roles in a community, accordingly, the topic proportions of them in a community are worthy of research. However, the topic proportions of several active authors in the same community are not explored by COCOMP and STC models.

3.4.3.3 The most participated community of an individual user

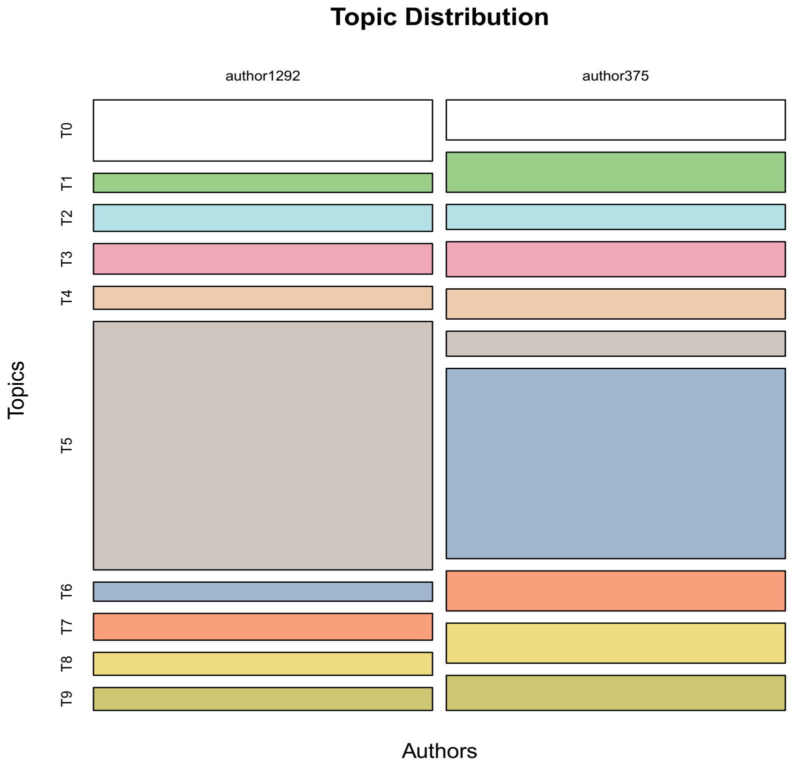
In our models, each user can belong to multiple communities. It is very useful to explore the information of a user's most participated community. In Table 3.9, we present the main authors, recipients, the major topics and their sentiment proportions in the most participated community (community 1) of the user *Steven Kean*.

We list the top 4 main topics with their probabilities in the parentheses. For topic 3 (development), the dominant sentiment is positive, while it is negative in topic 0 (office work). In addition, topic 2 (state power) and topic 3 (California energy) are also prevalent in this community. It is obvious from Table 3.9 that *Steven Kean* is an important person with a very high proportion 0.4355 in this community. Also *Jeff Dasovich* has high proportions in both author set and recipient set. *Richard Shapiro* is a vice president who is responsible for regulatory affairs, and *Drew Fossum* is also a vice president. It supports the view that these people are leading managers of the corporation and associated with important topics, like state power and California energy.

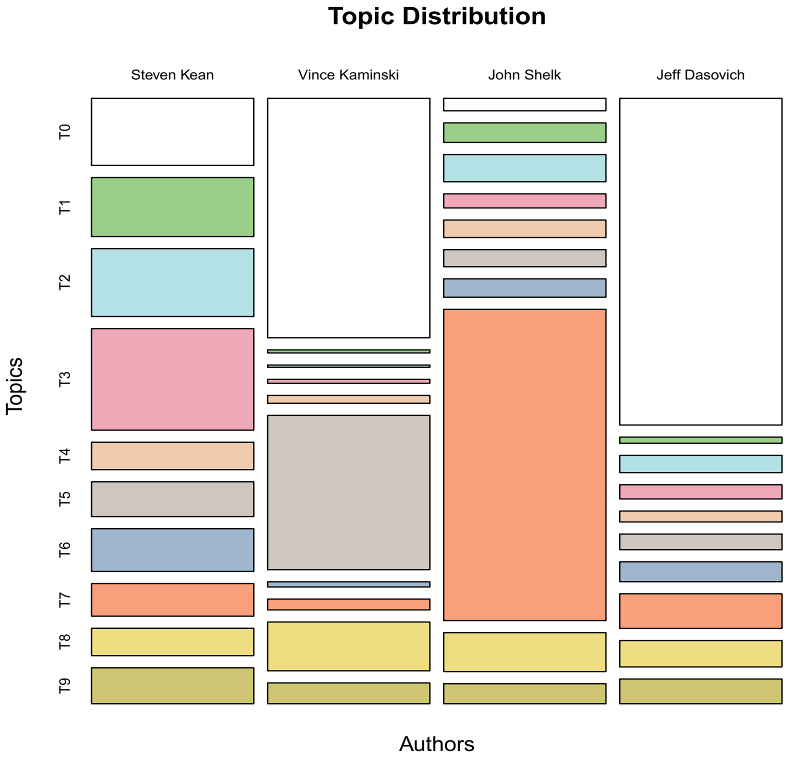
Unlike COCOMP and STC, our ASTC and ASTCx models can find both the main authors and main recipients in the same community. Typically, the information of these authors and recipients can help us further understand the whole community.

Table 3.9: Selected information for the most participated community (community 1) of Steven Kean (steven.kean@enron.com), (M=10, K=10).

Topic	Sentiment			People (denoted by the username of the enron email address)	
	Positive	Negative	Neutral	Main Authors	Main Recipients
3 (0.2046)	0.6749	0.2127	0.1124	steven.kean(0.4355), john.shelk(0.0677), ray.alvarez(0.0371), drew.fossum(0.0340), susan.lopez(0.0340), paul.simons(0.0340), j.kaminski(0.0156), jeff.dasovich(0.0034)	richard.shapiro, linda.robertson, jeff.dasovich
2 (0.1367)	0.2297	0.4018	0.3685		
0 (0.1350)	0.1407	0.6369	0.2224		
1 (0.1189)	0.9087	0.0460	0.0453		



(a) Topic distribution for author1292 and author375 in community 10 on Twitter-x dataset, M=12.



(b) Topic distribution for four authors in community 1 on Enron dataset, M=10.

Figure 3.17: Topic distribution of active authors in their communities, $K = 10$.

3.4.3.4 Sentiment distribution of topics

It is interesting to study the sentiment distribution of some authors' topics. The sentiment proportions of topics for some randomly selected authors are illustrated in Figure 3.18. As can be seen from Figure 3.18(a) that the dominant sentiment for topic 7, 8 and 9 is sentiment 2(neutral), while the main sentiments for other topics are either positive or negative. For the Enron dataset (Figure 3.18(b)), the author 1 in community 1 nearly has dominant sentiment towards all topics.

These results indicate that in each community, an author's sentiments towards different topics are various. Furthermore, the dominant sentiment incline for each topic of an author is available by our ASTC and ASTCx models. However, in COCOMP and STC, the sentiment distribution for each topic of the corresponding author in a community is not studied.

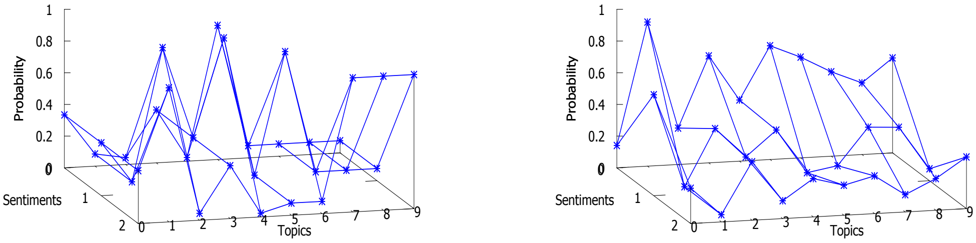
3.4.4 Comparing with Baseline Models

COCOMP model [125] and STC model [108] are used as the baselines. STC is an up-to-date community discovery model based on the combination of social links, topic and sentiment information in social networks, while COCOMP is a topic-level community discovery model in which the sentiment information is not considered. The results comparison between STC and COCOMP is aforementioned in Section 3.2.6.

In STC, ASTC and ASTCx, the sentiment information is studied in the model training. The differences between our STC model and our two new models, ASTC and ASTCx, can be summarised as follows. 1) STC model only considers the existence of users in each document, while our ASTC and ASTCx models also depict the author-recipient relationship in each document. 2) For each document, the topic z is drawn based on the topic distribution of the corresponding author a in the community c in our ASTC and ASTCx models, while the topic z is only drawn based on generic topic distribution in the community c by STC model. 3) Similarly, in our ASTC and ASTCx models, the sentiment assignment l is determined by the community assignment c, author a, and topic z together, which is more reasonable than STC model. 4) We model the recipient r based on community c and author a simultaneously in our ASTC and ASTCx models. 5) To better study the sentiment topics, the topic words and sentiment words are drawn separately in our ASTCx models.

In Figure 3.16, the topic distribution of a certain author #81 from its involved communities are illustrated, the detailed analysis has been reported in the previous subsection. In addition, ASTCx model can help us to observe the active authors' topic distributions in their communities shown in Figure 3.17. Besides the topic distribution of authors, Figure 3.18 shows the authors' sentiment distributions in their corresponding communities by our ASTCx model, which enables us to further understand the valuable information of members of communities, which is not accessible by STC model.

Table 3.9 lists the sentiment topics and main people in the most participated community of user *Steven Kean*. Notice that both the main authors and main recipients are listed, which are all the core members in the community. However, the main author and recipient information cannot be directly got by STC model.



(a) Sentiment distribution of topics for author 33 in community 6 on Twitter-x dataset, M=12.

(b) Sentiment distribution of topics for author 1 in community 1 on Enron dataset, M=10.

Figure 3.18: Sentiment distribution of topics for individual authors within their communities, $K = 10$.

3.4.4.1 Perplexity results comparison for COCOMP, STC, ASTC and ASTCx models.

In this work, the perplexity results comparison for COCOMP, STC, ASTC and ASTCx models is conducted. The perplexity equation for our ASTC model is shown in Eq.3.13. The lower perplexity score, the better performance the model has. In Eq.3.13, D_{test} denotes the held-out testing documents, $\tilde{\mathbf{w}}_m$ is the words from testing documents appeared in community m, $\mathbf{w}$ represents the words in the training documents. n_m is the number of words in community m. In Eq.3.14, $n_m^{(t)}$ is the number of times a term t seen in community m, and c_{w_n} is the community that the word w_n appears in. a_{w_n} is the author that the word w_n appears in his/her documents.

Different from ASTC model, the topic words and sentiment words are treated differently, the equation of perplexity for ASTCx model is represented in Eq.3.16. $\tilde{\mathbf{w}}'_m$ denotes the sentiment words from testing documents in community m, $\tilde{\mathbf{w}}''_m$ represents the topic words from testing documents in community m. In Eq.3.17, $c_{w'_n}$ is the community that the sentiment word w'_n appears in, $a_{w'_n}$ is the author that the sentiment word w'_n appears in his/her documents. $n_m'^{(t)}$ denotes the number of times a sentiment term t appears in community m. In Eq.3.19, $c_{w''_n}$ is the community that the topic word w''_n appears in. In addition, $a_{w''_n}$ is the author that the topic word w''_n appears in his/her documents, and $n''^{(t)}_m$ denotes the number of times a topic term t seen in community m.

The details of the perplexity equations for the STC model are shown in Section 3.2.6.3, and the perplexity equations for the COCOMP model can be seen in the work [125].

$$Perplexity_{\_ASTC}(D_{test}) = \exp\left\{-\frac{\sum_{m=1}^{M}\log Pro(\tilde{\mathbf{w}}_m|\mathbf{w})}{\sum_{m=1}^{M} n_m}\right\} \tag{3.13}$$

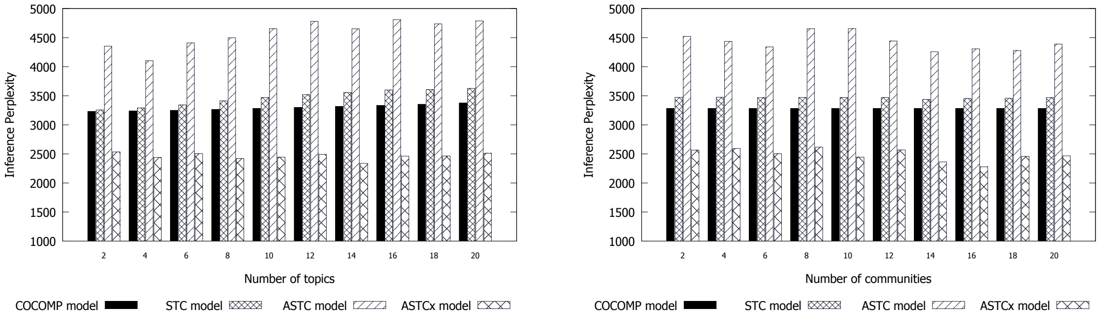
$$\begin{aligned}
&Pro(\tilde{\mathbf{w}}_m|\mathbf{w}) \\
&= \prod_{n=1}^{n_m}\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S} Pro(w_n = t|z_n = k, l_n = s) \times Pro(l_n = s|z_n = k, c_{w_n} = m, a_{w_n} = p) \\
&\quad \times Pro(z_n = k|c_{w_n} = m, a_{w_n} = p) \times Pro(a_{w_n} = p|c_{w_n} = m) \\
&= \prod_{t=1}^{V}\left(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S}\phi_{k,s,t}\pi_{m,p,k,s}\theta_{m,p,k}\lambda^a_{m,p}\right)^{n_m^{(t)}}
\end{aligned} \tag{3.14}$$

$$\log Pro(\tilde{\mathbf{w}}_m|\mathbf{w}) = \sum_{t=1}^{V} n_m^{(t)} \log(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S}\phi_{k,s,t}\pi_{m,p,k,s}\theta_{m,p,k}\lambda^a_{m,p}) \tag{3.15}$$

$$Perplexity_{\_ASTCx}(D_{test}) = \exp\left\{-\frac{\sum_{m=1}^{M}(\log Pro(\tilde{\mathbf{w}}'_m|\mathbf{w}) + \log Pro(\tilde{\mathbf{w}}''_m|\mathbf{w}))}{\sum_{m=1}^{M} n_m}\right\}. \tag{3.16}$$

$$\begin{aligned}
&Pro(\tilde{\mathbf{w}}'_m|\mathbf{w}) \\
&= \prod_{n=1}^{n'_m}\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S} Pro(w'_n = t|z_n = k, l_n = s) \times Pro(l_n = s|z_n = k, c_{w'_n} = m, a_{w'_n} = p) \\
&\quad \times Pro(z_n = k|c_{w'_n} = m, a_{w'_n} = p) \times Pro(a_{w'_n} = p|c_{w'_n} = m) \\
&= \prod_{t=1}^{V'}\left(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S}\phi'_{k,s,t}\pi_{m,p,k,s}\theta_{m,p,k}\lambda^a_{m,p}\right)^{n'^{(t)}_m}
\end{aligned} \tag{3.17}$$

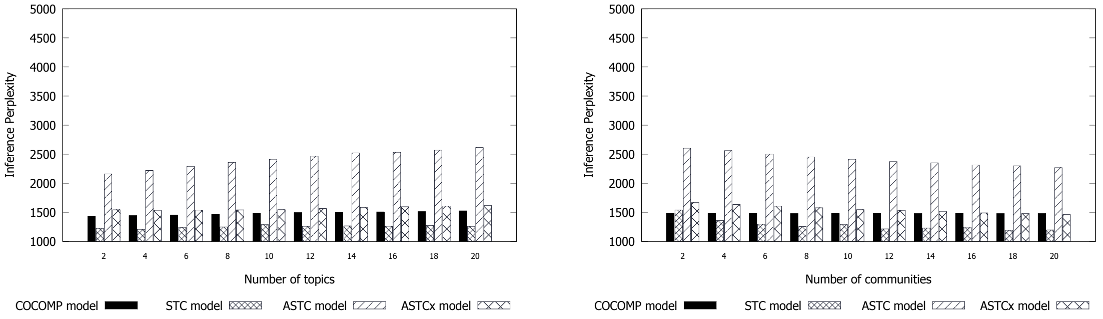
$$\log Pro(\tilde{\mathbf{w}}'_m|\mathbf{w}) = \sum_{t=1}^{V'} n'^{(t)}_m \log(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\sum_{s=1}^{S}\phi'_{k,s,t}\pi_{m,p,k,s}\theta_{m,p,k}\lambda^a_{m,p}) \tag{3.18}$$



(a) Perplexity under varying number of topics, $M = 10$.

(b) Perplexity under varying number of communities, $K = 10$.

Figure 3.19: Perplexity results comparison for COCOMP, STC, ASTC and ASTCx models on Enron dataset.



(a) Perplexity under varying number of topics, $M = 10$.

(b) Perplexity under varying number of communities, $K = 10$.

Figure 3.20: Perplexity results comparison for COCOMP, STC, ASTC and ASTCx models on Twitter-x dataset.

$$\begin{aligned} Pro(\tilde{\mathbf{w}}''_m|\mathbf{w}) = &\prod_{n=1}^{n''_m}\sum_{p=1}^{P_a}\sum_{k=1}^{K} Pro(w''_n = t|z_n = k) \times Pro(z_n = k|c_{w''_n} = m, a_{w''_n} = p) \\ &\times Pro(a_{w''_n} = p|c_{w''_n} = m) \qquad (3.19) \\ = &\prod_{t=1}^{V''}\left(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\phi''_{k,t}\theta_{m,p,k}\lambda^a_{m,p}\right)^{n''^{(t)}_m} \end{aligned}$$

$$\log Pro(\tilde{\mathbf{w}}''_m|\mathbf{w}) = \sum_{t=1}^{V''} n''^{(t)}_m \log(\sum_{p=1}^{P_a}\sum_{k=1}^{K}\phi''_{k,t}\theta_{m,p,k}\lambda^a_{m,p}) \qquad (3.20)$$

Figure 3.19(a) and Figure 3.19(b) show the comparison of perplexity for COCOMP, STC, ASTC and ASTCx models under varying number of topics and communities on Enron dataset. As we can see from Figure 3.19(a) and Figure 3.19(b), ASTCx model has the lower perplexity value than other three models on Enron dataset.

The perplexity comparisons on Twitter-x dataset for COCOMP, STC, ASTC and ASTCx models under varying number of topics and communities are given in Figure 3.20(a) and Figure 3.20(b). It is clear that ASTCx model is superior to ASTC model with lower perplexity. In comparison with COCOMP and STC model on Twitter-x dataset, ASTCx is not the best in terms of perplexity, however, it is close to COCOMP and STC models.

Overall, ASTCx has the comparable perplexity value to the COCOMP, STC and ASTC models.

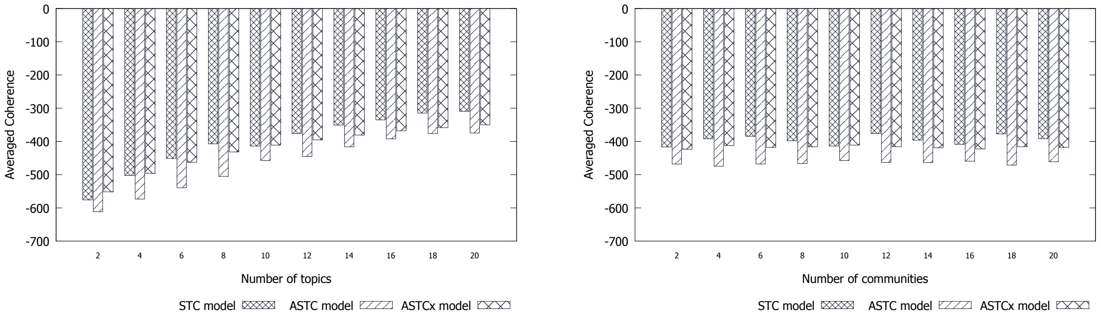
3.4.4.2 Topic-sentiment coherence scores for STC, ASTC and ASTCx models.

Topic coherence is another metric for assessing topic models. The Intrinsic UMass topic coherence measure [69] computes the topic coherence scores based on word co-occurrence statistics from the documents without external corpus. The topic coherence can be defined as $Coherence = \sum_{i<j} S_{UMass}(w_i, w_j)$, where $S_{UMass}(w_i, w_j) = \log \frac{N(w_i,w_j)+1}{N(w_i)}$ is an asymmetric pairwise score function, $N(w_i)$ denotes the count of documents containing the word w_i, $N(w_i, w_j)$ is the count of documents containing both word w_i and word w_j, here the word w_i and word w_j are from the word list for a topic.

In our work, both topic and sentiment are considered in community modelling, however, in COCOMP model, the important sentiment information is not studied. Therefore, STC model is not included in this comparison. To further evaluate the performance of our STC, ASTC and ASTCx models, we compute topic-sentiment coherence scores in the light of above mentioned Intrinsic UMass topic coherence measure [69]. In our models, each sentiment topic is represented by positive topic words, negative topic words and neutral topic words. So, for each sentiment topic, the topic-sentiment coherence score is the average of three different coherence scores based on positive, negative and neutral topic words. The overall topic-sentiment coherence score is computed by averaging the coherence scores of all the topics for each model. The higher coherence score, the better performance of the model has.

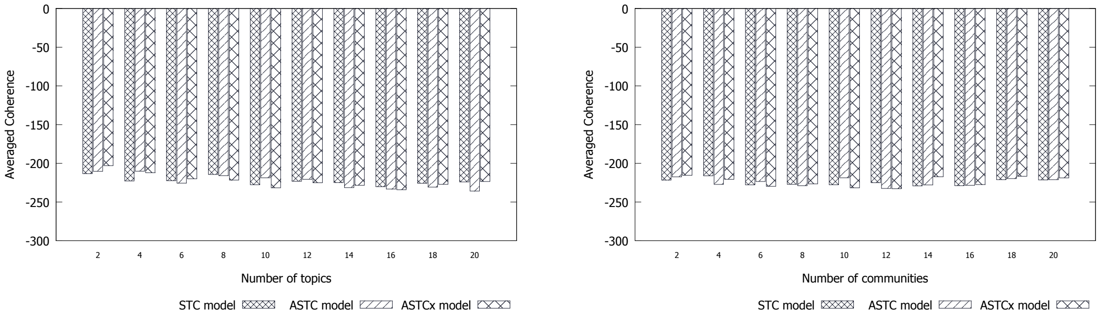
The comparisons of topic-sentiment coherence results for STC, ASTC and ASTCx models are illustrated in Figure 3.21 and Figure 3.22 on Twitter-x and Enron datasets respectively.

It can be seen from Figure 3.21(a) and Figure 3.21(b) that ASTCx model performs better with larger coherence scores than ASTC model on Enron dataset under varying number of topics and communities. By contrast, the averaged coherence scores between ASTCx and STC are very



(a) Coherence under varying K, $M = 10$. (b) Coherence under varying M, $K = 10$.

Figure 3.21: Topic-sentiment coherence scores on Twitter-x dataset.



(a) Coherence under varying K, $M = 10$. (b) Coherence under varying M, $K = 10$.

Figure 3.22: Topic-sentiment coherence scores on Enron dataset.

close. Overall, our ASTCx model is competitive among the three models in terms of averaged coherence on Enron dataset. Figure 3.22(a) and Figure 3.22(b) demonstrates that the averaged topic-sentiment coherence scores of ASTCx, ASTC and STC models are very similar on Enron dataset, which indicates that our ASTCx model is comparable to ASTC and STC models.

3.4.5 ASTCx vs. ASTC on the Identified Sentiment Topics

In this subsection, we report the sentiment topics identified by our ASTC and ASTCx models. In the ASTC model, each generated word follows a topic-sentiment distribution. We present selected topics obtained by the ASTC model on the Enron dataset in Table 3.11. Here we assume there are 10 topics and 10 communities. For each topic, there are three possible sentiment polarities, i.e., positive, negative and neutral. Ten representative words with high probabilities are shown for each sentiment. In Table 3.11, we list the topic sentiment words of topic 1 (business) and topic 9 (California energy) for three sentiment polarities.

As stated in the previous sections, ASTCx model is an extension of the ASTC model, which

Table 3.10: The high-ranking topic words for selected topics obtained by ASTCx model on two datasets.

(a) High-ranking topic words on Twitter-x dataset, M=10, K=4.

Topic 0	Topic 1	Topic 2	Topic 3
apple	twitter	microsoft	google
iphone	ipad	app	android
siri	hey	windows	phone
store	wow	ipod	samsung
ios	support	ballmer	facebook
jobs	check	users	update
icloud	send	memories	tech
service	music	cloud	nexus
video	network	office	apps
upgrade	issues	post	amazon

(b) High-ranking topic words on Enron dataset, M=10, K=10.

Topic 0	Topic 2	Topic 4	Topic 7
message	power	business	draft
forwarded	state	term	john
richard	energy	services	issues
corp	california	questions	meeting
vince	electricity	party	comments
office	utilities	process	linda
kaminski	gas	deal	discussion
meeting	prices	relationship	call
dasovich	davis	employees	steve
email	contracts	manager	committee

divides the words into topic words and sentiment words before the model construction. For ASTCx model, the high-ranking topic words on Twitter-x and Enron are shown in Table 3.10, and the top-ranking sentiment words on the two datasets based on the topics in Table 3.10 are given in Table 3.12. For Twitter-x dataset, the selected topics are 0 (apple), 1 (twitter), 2 (microsoft) and 3 (google). And on Enron dataset, the labels for topic 0, 2, 4 and 7 are office work, state power, company business, and committee meeting, respectively. It is obvious from Table 3.10 and Table 3.12 that the topic words and sentiment words are represented separately by ASTCx model, which can provide better word description for sentiment topics than the mixed topic-sentiment words shown in Table 3.11 by ASTC model. Note that the topic sequences in ASTC and ASTCx are different, for instance, on Enron dataset, topic 4 in ASTCx and topic 1 in ASTC are the same.

Table 3.11: The high-ranking topic-sentiment words for selected topics obtained by ASTC model on Enron dataset, M=10, K=10

Topic 1			Topic 9		
Positive	Negative	Neutral	Positive	Negative	Neutral
business	crisis	determine	services	contracts	california
office	long	decide	stock	long	energy
information	problem	shanna	business	yesterday	davis
issue	cut	reasons	term	megawatt	president
regulatory	avoid	facts	million	crisis	angeles
employees	vice	imagine	organization	thurs	state
support	hard	touch	great	settlement	news
good	low	houston	equity	refunds	words
study	failure	afternoon	forward	diego	price
account	year	times	people	association	april

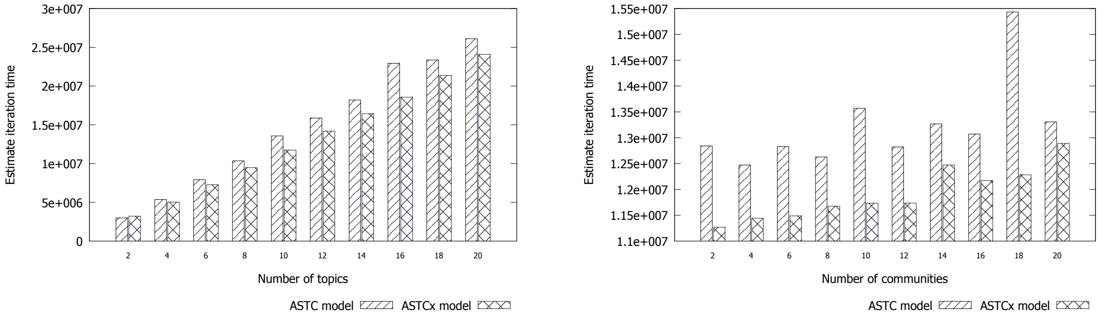
Table 3.12: The high-ranking sentiment words for selected topics obtained by ASTCx model on two datasets.

(a) High-ranking sentiment words on Twitter-x dataset, M=10, K=4.

Topic 0			Topic 1			Topic 2			Topic 3		
Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral
free	waiting	finally	great	lost	tomorrow	good	world	today	dear	sold	fast
amazing	bad	coming	glad	digital	apparently	awesome	fucking	big	nice	powered	stuck
pretty	changed	posted	interesting	infinite	blue	top	called	full	working	total	literally
simple	disappointed	tonight	real	side	huge	live	angry	home	loving	announced	prime
yesterday	spotted	ordered	properly	slow	half	cool	black	yeah	sweet	wrong	played
easy	terrible	similar	sharing	crazy	future	happy	sad	air	beautiful	missing	fully
incredible	hard	cutting	white	stupid	simply	funny	evil	mobile	smart	worst	knowing
official	dead	lined	perfect	annoyed	totally	social	anti	absolutely	light	worse	coming
easier	tired	documentary	promised	dark	addicted	open	killing	coming	principal	beat	ringing
impressive	turned	possibly	patiently	useless	hanging	important	loose	running	global	awful	included

(b) High-ranking sentiment words on Enron dataset, M=10, K=10.

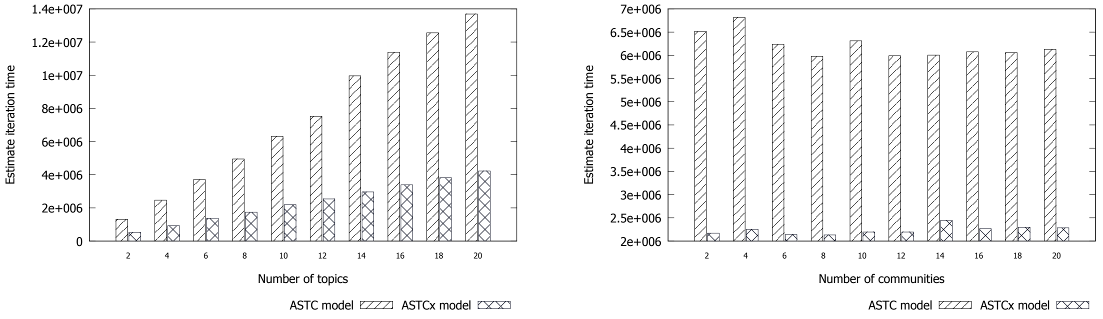
Topic 0			Topic 2			Topic 4			Topic 7		
Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral	Positive	Negative	Neutral
confidential	subject	complete	natural	long	billion	forward	stock	firm	attached	local	air
original	part	central	regulatory	small	high	pass	lost	red	august	commercial	pat
privileged	attached	familiar	independent	paid	major	strategic	capital	personal	original	long	large
forward	received	troubled	competitive	electric	federal	center	daily	select	legal	future	involved
revised	hearing	certified	light	public	utility	confidential	based	standard	significant	national	federal
joined	written	decided	open	pacific	yesterday	shared	low	covered	final	continent	talking
interesting	required	scheduled	responsible	based	reliant	developed	found	major	forward	cross	model
strategic	attended	needed	reasonable	federal	air	longer	underlying	combined	legislative	supporting	included
successful	conservative	substitute	worth	expected	renewable	corporate	booked	middle	clear	critical	specific
relevant	requested	pacific	based	proposed	wholesale	back	contingent	lowest	early	suspect	extended



(a) Running time under varying number of topics K, M=10.

(b) Running time under varying number of communities M, K=10.

Figure 3.23: Running time (*ms*) for ASTC and ASTCx on Twitter-x dataset.



(a) Running time under varying number of topics K, M=10.

(b) Running time under varying number of communities M, K=10.

Figure 3.24: Running time (*ms*) for ASTC and ASTCx on Enron dataset.

3.4.6 Running Time for ASTC and ASTCx.

For ASTC and ASTCx models, the main running time is cost in the Gibbs sampling process. Figure 3.23 and Figure 3.24 show the estimated iteration time for ASTC and ASTCx models versus different topic and community number settings. It is obvious from Figure 3.23(a) and Figure 3.23(b) that ASTCx is less time-consuming than ASTC model on Twitter-x dataset, which means that ASTCx is more efficient than ASTC model for community discovery. Similarly, Figure 3.24(a) and Figure 3.24(b) show that ASTCx model performs much better with less iteration time consuming in the model training than ASTC model on Enron dataset under varying number of topics K and communities M.

3.5 Summary

Discovering communities from networks has been widely studied in recent years, which can help us to understand the latent knowledge and distributions within them. In this chapter, we first propose a novel community discovery model, STC, to explore communities with different topic-sentiment distributions. This model is built by combining content, links and sentiment words seamlessly, which can identify communities in a level of sentiment analysis. While most of existing methods for community identification fail to consider the valuable sentiment factor in the networks. Evaluations on two types of real-world datasets show that our model can detect sentiment-level communities and can achieve comparable performance.

To address the weakness of STC model, we propose another two novel community discovery models, ASTC and ASTCx. Experimental results and analysis on two real-world datasets show that our models can effectively uncover communities with different distributions. According to the distributions in communities, we can find sentiment unambiguous communities with respect to certain topics. It might be applicable for the opinion analysis and decision making in business and marketing service.

Chapter 4

Expert User Learning in Question Answering Communities

The research on community question answering (CQA) has been paid increasing attention in recent years. In CQA, to reduce the number of unanswered questions and the time for askers to wait, it is very necessary to identify relevant experts or best answers for these questions. Generally, the experts' answers are more likely to be the best answers. Existing studies considered that user expertise is reflected by the voting scores of both answers and questions. However, voting scores of questions are not really related to user expertise. In this chapter, we first propose to depict users' expertise based on answers and their descriptive ability based on questions. Specifically, we present a novel probabilistic model, User Topical Ability Model (UTAM), to depict the topic-specific user ability, in which the textual information (words and tags) and voting scores of questions are combined to model the topical specific user descriptive ability, while the user topic-specific expertise is depicted by integrating the textual information (words and tags) and voting scores of answers. Apart from the intrinsic textual and voting information, we also explore the valuable social links within a QA community. To exploit social information in CQA, the link analysis is also considered. To be exact, we proposed a new method, User Social Topic Ability (USTA), by integrating the results of UTAM with the link structure to further model the ability of users. In many CQA services, like Stack Overflow, the tag information of each question is available, which can be viewed as the keywords of the question. Compared with the topic information, the tags are more informative. Motivated by this case, we also try to study the user expertise under tags. The extensive experiments on the large datasets from Stack Overflow service demonstrate that our models and methods can yield comparable or even better performance than the state-of-the-art models. It is worth noticing that our user-tag PMF based method can reduce the running time greatly.

Chapter outline

The rest of this chapter is organized as follows: Section 4.1 describes our proposed UTAM model. We present our USTA method by incorporating social links based on UTAM in Section 4.2. Section 4.3 presents the framework of our user-tag PMF based expert user recommendation method. In Section 4.4, to evaluate our UTAM model and USTA method, we show the experiments and performance analysis from three aspects on a large CQA dataset. In Section 4.5, we present the evaluation of our user-tag PMF based method and the baselines on a large dataset from Stack Overflow. At last, we summarize this chapter in Section 4.6.

4.1 Joint Modelling User Topical Expertise and Descriptive Ability (UTAM)

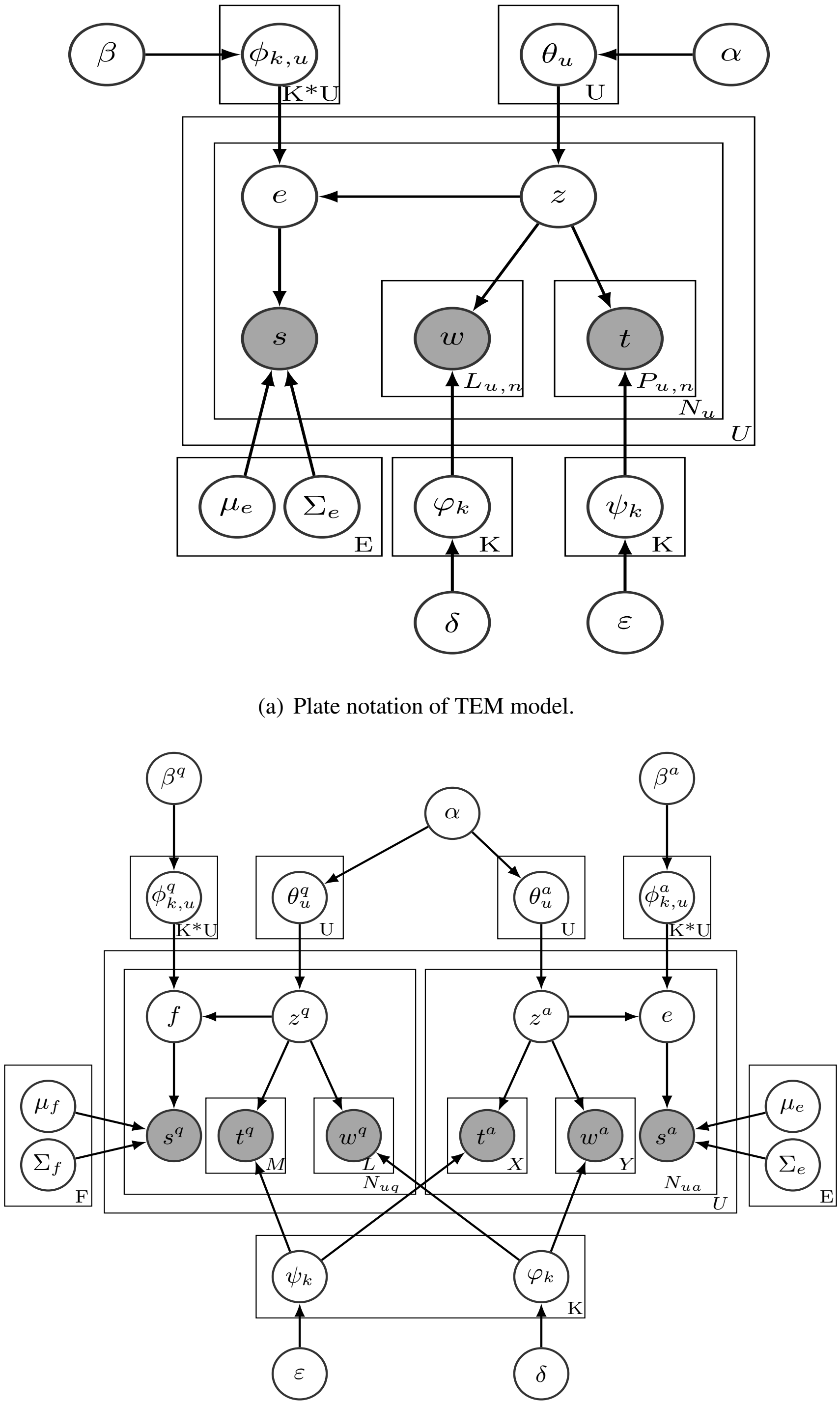
4.1.1 Model Framework

Our model is related to the Topic Expertise Model (TEM) in [110]. To give a clear comparison, we show the graphical notations of TEM and our UTAM in Figure 4.1(a) and Figure 4.1(b), respectively.

In our UTAM, we model the topical expertise and descriptive ability for each user. Specifically, we treat questions and answers of users as different types of posts, and we use N_{uq}, N_{ua} to denote the number of questions and answers of user u, respectively. In UTAM, there are four latent variables, namely, question topic z^q, descriptive ability level f, answer topic z^a and expertise level e, where f is defined under question topic z^q, and e is related to answer topic z^a. Due to the different implications between the votes of questions and answers, we learn the descriptive ability levels f based on the vote scores s^q of questions, and the expertise levels e are learnt based on the vote scores s^a of answers. As [110], we also assume that vote scores follow Gaussian distributions. The ability levels can be represented by the mean value of the Gaussian distributions, where a high user ability level is associated with a high mean value. For each question, the observable variables are content words w^q, tags t^q and voting score s^q, in addition, we consider content word w^a, tags t^a and voting score s^a as the observable variables for each answer. The tags of answers in general CQA sites are usually unavailable, as TEM [110], we also use the tags of their corresponding questions in our UTAM model.

The main difference between TEM and our UTAM can be summarised as follows:

- TEM considers that the voting scores of questions and answers are the reflection of users'



(a) Plate notation of TEM model.

(b) Plate notation of our UTAM model.

Figure 4.1: Graphical notation of models.

expertise, while our UTAM depicts users' expertise only based on the answers who previously provided.

- The topic distributions of questions and answers from each user are usually very different. Unlike TEM, our UTAM depicts user question topic s^q and answer topic z^a individually.

We use U, K, E to denote the number of user, number of topics and number of expertise levels in both TEM and our UTAM. In TEM, N_u is the number of Q&A posts of user u; $L_{u,n}$ and $P_{u,n}$ are used to represent the number of content words and number of tags for the n-th post of user u, respectively. In UTAM, F is the number of descriptive ability levels, M and L are the number of tags and content words in a question. And we use X and Y to denote the number of tags and content words in an answer.

The following are the details of parameters in our UTAM, where $Dir(\cdot)$ represents Dirichlet distribution.

- $\phi^q_{k,u}$: the descriptive ability level proportion of user u on topic k, which has a Dirichlet distribution with $\phi^q_{k,u}|\beta^q \sim Dir(\beta^q)$.

- $\phi^a_{k,u}$: the expertise level proportion of user u on topic k, which has a Dirichlet distribution with $\phi^a_{k,u}|\beta^a \sim Dir(\beta^a)$.

- θ^q_u: the topic proportion of user u based on questions, which follows a Dirichlet distribution with $\theta^q_u|\alpha \sim Dir(\alpha)$.

- θ^a_u: the topic proportion of user u based on answers, which follows a Dirichlet distribution with $\theta^a_u|\alpha \sim Dir(\alpha)$.

- ψ_k: the topic distribution over tags, $\psi_k|\epsilon \sim Dir(\epsilon)$.

- φ_k: the topic distribution over words, $\varphi_k|\delta \sim Dir(\delta)$.

- μ_e and Σ_e: the mean and precision of Gaussian distribution for expertise level e with Normal-Gamma distribution priors, $\mathcal{N}(\mu_e, \Sigma_e) \sim \mathcal{NG}(\alpha_0, \beta_0, \mu_0, \kappa_0)$.

- μ_f and Σ_f: the mean and precision of Gaussian distribution for descriptive ability level f with Normal-Gamma distribution priors, $\mathcal{N}(\mu_f, \Sigma_f) \sim \mathcal{NG}(\alpha_0, \beta_0, \mu_0, \kappa_0)$.

We show the generative process of question posts and answer posts of each user u, $(u = 1, 2, ..., U)$ as follows, where $Mult(\cdot)$ denotes multinomial distribution, and $\mathcal{N}(\cdot, \cdot)$ is Gaussian distribution.

- For each question post qp, ($qp = 1, 2, ..., N_{uq}$)
  - Sample a topic assignment z^q, $z^q \sim Mult(\theta_u^q)$.
  - Choose a descriptive ability level f, $f \sim Mult(\phi_{z^q,u}^q)$
  - Draw a vote score s^q, $s^q \sim \mathcal{N}(\mu_f, \Sigma_f)$
  - For the l-th word, ($l = 1, 2, ..., L$)
    - ∗ Draw a word assignment w^q, $w^q \sim Mult(\varphi_{z^q})$
  - For the m-th tag ($m = 1, 2, ..., M$)
    - ∗ Sample a tag assignment t^q, $t^q \sim Mult(\psi_{z^q})$
- For each answer post ap, ($ap = 1, 2, ..., N_{ua}$)
  - Sample a topic assignment z^a, $z^a \sim Mult(\theta_u^a)$.
  - Choose an expertise level e, $e \sim Mult(\phi_{z^a,u}^a)$
  - Draw a vote score s^a, $s^a \sim \mathcal{N}(\mu_e, \Sigma_e)$
  - For the y-th word, ($y = 1, 2, ..., Y$)
    - ∗ Choose a word assignment w^a, $w^a \sim Mult(\varphi_{z^a})$
  - For the x-th tag ($x = 1, 2, ..., X$)
    - ∗ Sample a tag assignment t^a, $t^a \sim Mult(\psi_{z^a})$

4.1.2 Inference and Parameter Estimation

The main statistics and symbols for the UTAM model inference are shown in Table 4.1, where we use $b = (u, qp)$ to denote the qp-th question post of user u, and $c = (u, ap)$ is used to denote the ap-th answer post of user u.

Table 4.1: The statistics and variables.

Statistic/Variable	**Description**
$n^q_{u,k}$ ($n^{q(-b)}_{u,k}$)	the number of questions of user u are assigned to topic k (excluding question b);
$n^a_{u,k}$ ($n^{a(-c)}_{u,k}$)	the number of answers of user u are assigned to topic k (excluding answer c);
$n^q_{k,u,f}$ ($n^{q(-b)}_{k,u,f}$)	the number of questions belonging to user u assigned to topic k with descriptive ability level f(excluding question b);
$n^a_{k,u,e}$ ($n^{a(-c)}_{k,u,e}$)	the number of answers belonging to user u assigned to topic k with expertise level e(excluding answer c);
$n^q_{k,v}$ ($n^{q(-b)}_{k,v}$)	the number of word v in questions(excluding question b) assigned to topic k;
$n^a_{k,v}$ ($n^{a(-c)}_{k,v}$)	the number of word v in answers(excluding answer c) assigned to topic k;
$n^q_{k,t}$ ($n^{q(-b)}_{k,t}$)	the number of tag t in questions(excluding question b) assigned to topic k;
$n^a_{k,t}$ ($n^{a(-c)}_{k,t}$)	the number of tag t in answers(excluding answer c) assigned to topic k;
$h_{b,v}$	the number of word v in question b;
$h_{c,v}$	the number of word v in answer c;
h_b	the number of words in question b;
h_c	the number of words in answer c;
$g_{b,t}$	the number of tag t in question b;
$g_{c,t}$	the number of tag t for answer c;
g_b	the number of tags in question b;
g_c	the number of tags for answer c;
$\mathbf{v}_b$	the word set of question b;
$\mathbf{v}_c$	the word set of answer c;
$\mathbf{t}_b$	the tag set of question b;
$\mathbf{t}_c$	the tag set of answer c.

The posterior distribution of z, f, and e can be written as follows:

$$
\begin{aligned}
&P(z^q_b = k, f_b = f | \mathbf{z}^q_{-b}, \mathbf{z}^a, \mathbf{w}^q, \mathbf{w}^a, \mathbf{f}_{-b}, \mathbf{s}^q, \mathbf{t}^q, \mathbf{t}^a) \\
&\propto \frac{\prod_{v \in \mathbf{v}_b} \prod_{i=0}^{h_{b,v}-1} (\delta_v + n^{q(-b)}_{k,v} + n^a_{k,v} + i)}{\prod_{i=0}^{h_b-1} (\sum_{v=1}^{V} \delta_v + n^{q(-b)}_{k,v} + n^a_{k,v} + i)} \\
&\cdot \frac{n^{q(-b)}_{k,u,f} + \beta^q_f}{\sum_{f=1}^{F} n^{q(-b)}_{k,u,f} + \beta^q_f} \cdot \mathcal{N}(s^q_b | \mu_f, \Sigma_f) \\
&\cdot \frac{\prod_{t \in \mathbf{t}_b} \prod_{p=0}^{g_{b,t}-1} (\varepsilon_t + n^{q(-b)}_{k,t} + n^a_{k,t} + p)}{\prod_{p=0}^{g_b-1} (\sum_{t=1}^{T} \varepsilon_t + n^{q(-b)}_{k,t} + n^a_{k,t} + p)} \\
&\cdot \frac{n^{q(-b)}_{u,k} + \alpha_k}{\sum_{k=1}^{K} n^{q(-b)}_{u,k} + \alpha_k}
\end{aligned}
\tag{4.1}
$$

$$\begin{aligned}
&P(z_c^a = k, e_c = e|\mathbf{z}_{-c}^a, \mathbf{z}^q, \mathbf{w}^q, \mathbf{w}^a, \mathbf{e}_{-c}, \mathbf{s}^a, \mathbf{t}^q, \mathbf{t}^a) \\
&\propto \frac{\prod_{v\in\mathbf{v}_c}\prod_{i=0}^{h_{c,v}-1}(\delta_v + n_{k,v}^{a(-c)} + n_{k,v}^q + i)}{\prod_{i=0}^{h_c-1}(\sum_{v=1}^{V}\delta_v + n_{k,v}^{a(-c)} + n_{k,v}^q + i)} \\
&\cdot \frac{n_{k,u,e}^{a(-c)} + \beta_e^a}{\sum_{e=1}^{E} n_{k,u,e}^{a(-c)} + \beta_e^a} \cdot \mathcal{N}(s_c^a|\mu_e, \Sigma_e) \\
&\cdot \frac{\prod_{t\in\mathbf{t}_c}\prod_{p=0}^{g_{c,t}-1}(\varepsilon_t + n_{k,t}^{a(-c)} + n_{k,t}^q + p)}{\prod_{p=0}^{g_c-1}(\sum_{t=1}^{T}\varepsilon_t + n_{k,t}^{a(-c)} + n_{k,t}^q + p)} \\
&\cdot \frac{n_{u,k}^{a(-c)} + \alpha_k}{\sum_{k=1}^{K} n_{u,k}^{a(-c)} + \alpha_k}
\end{aligned} \tag{4.2}$$

The parameters, $\theta_{u,k}^q$, $\theta_{u,k}^a$, $\phi_{k,u,f}^q$, $\phi_{k,u,e}^a$, $\varphi_{k,v}$ and $\psi_{k,t}$ can be updated as follows:

$$\theta_{u,k}^q = \frac{n_{u,k}^q + \alpha_k}{\sum_{k=1}^{K} n_{u,k}^q + K\alpha_k}$$

$$\theta_{u,k}^a = \frac{n_{u,k}^a + \alpha_k}{\sum_{k=1}^{K} n_{u,k}^a + K\alpha_k}$$

$$\phi_{k,u,f}^q = \frac{n_{k,u,f}^q + \beta_f^q}{\sum_{f=1}^{F} n_{k,u,f}^q + F\beta_f^q}$$

$$\phi_{k,u,e}^a = \frac{n_{k,u,e}^a + \beta_e^a}{\sum_{e=1}^{E} n_{k,u,e}^a + E\beta_e^a}$$

$$\varphi_{k,v} = \frac{n_{k,v}^q + n_{k,v}^a + \delta_v}{\sum_{v=1}^{V} n_{k,v}^q + n_{k,v}^a + V\delta_v}$$

$$\psi_{k,t} = \frac{n_{k,t}^q + n_{k,t}^a + \varepsilon_t}{\sum_{t=1}^{T} n_{k,t}^q + n_{k,t}^a + T\varepsilon_t}$$

where V and T denote the size of word vocabulary and that of tag vocabulary respectively. The parameters μ_e, Σ_e for expert level e, and μ_f, Σ_f for descriptive ability f are updated as the way used in [110].

4.2 USTA: User Social Topical Ability

CQARank [110] is a framework for recommendation analysis in community question answering, which is an extension of PageRank [76] algorithm by incorporating user network structure from Q&A graph into the results of the Topic Expertise Model (TEM) [110]. In CQARank, it is assumed that users who answer the question of high expertise users are more likely to have high expertise as well. Here the TEM model is a generative model to learn user topics and expertise by using the content and voting information of Q&A posts. The description of TEM model is given in Section 4.1.1.

For our new method USTA, we also consider the social link structures in CQA for further study of user topical ability. In QA user networks, the social links are built based on the question answering behavior among users. We assume that a user who provides answers to an expert user is more likely to have high level ability as well. In CQA, we use $\mathbb{G} = (\mathbb{V}, \mathbb{E}, \mathbb{O})$ to denote a weighted directed user interaction graph, where $\mathbb{V}$ represents a set of all users, $\mathbb{E}$ denotes a collection of directed edges between users, and $\mathbb{O}$ is a collection of edge weights. Edge (u_i, u_j) means user u_j answered the questions of user u_i. The more answers user u_j provide to user u_i, the higher the weight $o_{i,j}$ is. Generally, one tends to answer questions of others when they share similar topics. We measure the topical similarity $sim_z(i \to j)$ between asker u_i and answerer u_j under topic z as follows.

$$sim_z(i \to j) = 1 - |\theta^q_{i,z} - \theta^a_{j,z}|$$

where $\theta^q_{i,z}$ is the question topic distribution of user u_i while $\theta^a_{j,z}$ is the answer topic distribution of user u_j in UTAM.

Referring to CQARank [110], we compute the transition probability $P_z(i \to j)$ of a random surfer from u_i to u_j on CQA graph $\mathbb{G}$.

$$\begin{aligned} & P_z(i \to j) \\ & = \begin{cases} \frac{o_{i,j} \cdot sim_z(i \to j)}{\sum_{r=1}^{|\mathbb{V}|} o_{i,r} \cdot sim_z(i \to r)}, & \text{if} \sum_r o_{i,r} \neq 0. \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{4.3}$$

We compute the topical descriptive ability saliency score $SS^f_z(u_i)$ and topical expertise saliency score $SS^e_z(u_i)$ of u_i in Eq.4.4 and Eq.4.5 respectively.

In USTA, To make use of the Q&A link structures and further estimate each user's expertise score and descriptive ability score under topics, we define two saliency scores: 1) Topical descriptive ability saliency score $SS^f_z(u_i)$ of user u_i under topic z, which aggregates the results from U-TAM and user link analysis to measure the final descriptive ability score. $\sum\limits_{j:u_j \to u_i} SS^f_z(u_j) \cdot P_z(j \to i)$ in Eq.4.4 shows the consideration of user link analysis from Q&A graph; 2) Topical expertise saliency score $SS^e_z(u_i)$ of user u_i under topic z. The definition of $SS^e_z(u_i)$ can be presented as Eq.4.5, which also combines the results from UTAM and user link analysis to estimate the final expertise ability score.

The two saliency scores $SS^f_z(u_i)$ and $SS^e_z(u_i)$ shown in Eq.4.4 and Eq.4.5 are represented in a recursive way, in which both the topical interests and ability are considered together. $\lambda \in [0, 1]$

is a damping factor, $\theta^q_{u_i,z}$, $\theta^a_{u_i,z}$, $\phi^q_{z,u_i,f}$, $\phi^a_{z,u_i,e}$, μ_f and μ_e are the estimated parameters learned from our UTAM model.

$$\begin{aligned} SS^f_z(u_i) = \lambda \sum_{j:u_j \to u_i} & SS^f_z(u_j) \cdot P_z(j \to i) \\ + (1-\lambda) \cdot \theta^q_{u_i,z} \cdot & \sum_{f=1}^{F} \phi^q_{z,u_i,f} \cdot \mu_f \end{aligned} \tag{4.4}$$

$$\begin{aligned} SS^e_z(u_i) = \lambda \sum_{j:u_j \to u_i} & SS^e_z(u_j) \cdot P_z(j \to i) \\ + (1-\lambda) \cdot \theta^a_{u_i,z} \cdot & \sum_{e=1}^{E} \phi^a_{z,u_i,e} \cdot \mu_e \end{aligned} \tag{4.5}$$

Both USTA and CQARank consider to incorporate the user network information from user Q&A graph into the results of their corresponding basic models. However, there are several differences between USTA and CQARank.

1) The similarity between user u_i and u_j under topic z, $sim_z(i \to j)$ is different. In CQARank, $sim_z(i \to j)_{CQARank} = 1 - |\theta'_{i,z} - \theta'_{j,z}|$, where θ' is row normalized matrix as user specific topic distribution in TEM, $\theta'_{i,z}$ and $\theta'_{j,z}$ denote the user specific topic distributions of user u_i and u_j under topic z, respectively. While in USTA, $sim_z(i \to j)_{USTA} = 1 - |\theta^q_{i,z} - \theta^a_{j,z}|$, here $\theta^q_{i,z}$ is the question topic distribution of user u_i and $\theta^a_{j,z}$ is the answer topic distribution of user u_j under topic z in UTAM.

2) In our USTA, we define two different saliency scores, user topic descriptive ability saliency score and topic expertise saliency score. Because user descriptive ability and expertise are two different abilities, so their corresponding saliency scores are defined separately. However, in CQARank, the saliency score is used for measuring the user topical expertise. In addition, the definitions of topical expertise in CQARank and USTA are different. The important descriptive ability is treated as expertise in CQARank.

4.3 User-Tag Modelling for Expert User Recommendation

In this section, we will introduce our tag-based framework for expert recommendation in community question answering. We first build user-tag expertise matrix based on the training dataset, then the latent user feature matrix and tag feature matrix will be obtained via user-tag matrix factorization. Finally we conduct expert recommendation based on the predicted expertise scores.

Table 4.2: A sample of user-tag score pairs for a given question.

Tag Id User Id	T_1	T_2	T_3	T_4
U_1	0.25	0.05	0.70	0.50
U_2	0.35	0.45	0.25	0.05
U_3	0.05	0.25	0.15	0.80

An example for ranking 3 candidate users based on 4 tags of a new question is shown in Table 4.2. It can be easily computed that the averaged score for user U_1 towards all the 4 tags is (0.25+0.05+0.70+0.50)/4=0.375. And such for U_2 and U_3 are 0.275 and 0.3125 respectively. Then we could get the ranking of users as U_1, U_3, U_2 in descending order.

4.3.1 User-Tag Matrix Factorization

In some CQA services, like Stack Overflow, each question has been attached several tags by askers, these tags are more representative than the content of questions. We assume that a user with high expertise regarding a question is likely to be skilled at its tags. Therefore, we learn the user expertise under tags, and we create the user-tag expertise matrix based on the user previously answered questions. In the user-tag expertise matrix, suppose there are M users and N tags, s_{ij} represents the averaged expertise score of user i for tag j. $U \in \mathbb{R}^{L\times M}$ and $Y \in \mathbb{R}^{L\times N}$ are latent user and tag L-dimensional feature matrices.

As PMF [90], the distribution of user U and tag Y can be shown in Eq.4.6, where $\mathcal{N}(\cdot,\cdot)$ denotes Gaussian distribution, U_i and Y_j are the user feature vector and tag feature vector, respectively.

$$\begin{aligned} P(U|\sigma_U^2) &= \prod_{i=1}^{M} \mathcal{N}(U_i|\mathbf{0}, \sigma_U^2\mathbf{I}), \\ P(Y|\sigma_Y^2) &= \prod_{j=1}^{N} \mathcal{N}(Y_j|\mathbf{0}, \sigma_Y^2\mathbf{I}) \end{aligned} \tag{4.6}$$

To learn U and Y, we also conduct the minimization of the following objective function.

$$\begin{aligned} L(S, U, Y) &= \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} \mathbb{I}_{ij}^{S}(s_{ij} - U_i^T Y_j)^2 \\ + \frac{\lambda_U}{2}\sum_{i=1}^{M} \|U_i\|_{\mathcal{F}}^2 &+ \frac{\lambda_Y}{2}\sum_{j=1}^{N} \|Y_j\|_{\mathcal{F}}^2 \end{aligned} \tag{4.7}$$

where $\|\cdot\|_{\mathcal{F}}$ represents the Frobenius norm, $\lambda_U = \sigma_S^2/\sigma_U^2$, and $\lambda_Y = \sigma_S^2/\sigma_Y^2$.

In Eq.4.7, $\mathbb{I}_{ij}^S$ is the indicator function, which can be described as Eq.4.8. We use $f(s) = \frac{s - s_{\min}}{s_{\max} - s_{\min}}$ to bound each rating score s within [0,1] before training, where $s_{\max}$ and $s_{\min}$ are the maximum and minimum ratings in the dataset.

$$\mathbb{I}_{ij}^S = \begin{cases} 1, & \text{if user i answered questions having tag } j. \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

We compute the local minimum of the above objective function $L(S, U, Y)$ by conducting gradient descent in U_i and Y_j, which are shown in Eq.4.9 and Eq.4.10.

$$\frac{\partial L}{\partial U_i} = \sum_{j=1}^{N} \mathbb{I}_{ij}^S (U_i^T Y_j - s_{ij}) Y_j + \lambda_U U_i \tag{4.9}$$

$$\frac{\partial L}{\partial Y_j} = \sum_{i=1}^{M} \mathbb{I}_{ij}^S (U_i^T Y_j - s_{ij}) U_i + \lambda_Y Y_j \tag{4.10}$$

4.3.2 Expert Recommendation

Once we get the user matrix U and tag matrix Y, the missing expertise value s_{ij} in S can be predicted by using $U_i^T Y_j$, then we can conduct expert user recommendation with the help of the completed user-tag expertise matrix.

4.4 Experimental Analysis for UTAM and USTA Models

4.4.1 Experimental Setup

4.4.1.1 Data Set

The data set we used is from a large CQA site, Stack Overflow, which covers a wide range of computer programming related topics. Specifically, we use a Stack Overflow Data Dump spanning from August 2008 to August 2010, which is publicly available. For the training set, we collect posts between July 5th 2009 and October 5th 2009, then users who have more than 85 posts[1] are selected to make sure there are enough posts from each user to learn the topics and user abilities. The raw training set is composed of 96899 answers and 9135 questions belonging to 606 users. With regard to the testing dataset, we first select questions ranging from October 6th 2009 to July 6th 2010. From these selected questions, we then choose questions with more than 3 answers, where the provider of each answer is in the user list of training set. We conducted data preprocessing for the data sets including code snippets deletion, stop word removal, etc.

[1] Note that the value can be chosen randomly, we use 85 in this chapter.

4.4.1.2 Parameter Settings

The Gibbs sampling used in each model inference is executed 500 iterations. As [110], we also set κ_0 as 1, α_0 as 1, μ_0 as the average value of votes based on training data, and β_0 as the average distance between randomly sampled 500 votes. The damping factor for saliency scores is set as λ = 0.2. According to multiple experiments, we set topic number K as 10, the number of descriptive levels F as 10, and the number of expertise levels E as 10 as well. In addition, referring to [32], we set hyperparameters α = 50/K, δ = 0.01, ϵ = 0.001, β^q = β^a = β = 0.01.

4.4.2 Baselines

We compare our UTAM and USTA methods with the state-of-the-art CQARank algorithm and TEM model proposed in [110]. In the following subsections, we will report the experimental results and comparisons in three different tasks, namely, expert ranking, answer recommendation, and similar question retrieval for a new question.

The distinctions between our UTAM and USTA method and baselines include two aspects:

First, the way of user ability modelling is different from the baselines. In the baseline methods, the vote scores of both questions and answers of a user are used together to learn the expertise of users. In fact, the voting score of a question only indicates the readability and clearity of the question, which reflects the user's descriptive ability or written skills rather than expertise. Our UTAM and USTA treats the user ability as the synthesis of user expertise and descriptive ability, where the user expertise is learnt based on answer votes while the descriptive ability is reflected by question votes.

Another difference is that the topic distributions of questions and answers of each user are usually different, while in TEM the topic distribution are modeled based on all the Q&A posts.

Notice that each experiment is conducted 10 runs for each model and the averaged results in terms of each evaluation metric are reported in the following subsections.

4.4.3 Evaluation on Expert Recommendation for a Given Question

For a given question, the main goal of expert user recommendation is to work out a ranking list of users who are the potential answer providers with high ability (expertise and descriptive ability) under certain topics. The ground truth is the ranking list of users obtained by ranking their voting scores of answers in the test data.

For each question q and the corresponding user testing set $\mathcal{U}$, we calculate the recommendation score $RS(u, q)$ for each user $u \in \mathcal{U}$ towards a question q, which is shown in Eq.4.11. Based on the recommendation scores, we can get a ranking list of users in decreasing order.

$$\begin{aligned}
RS(u,q) &= Expertise(u,q) \cdot Descriptive(u,q) \\
&= (UTCount(u,q) \cdot \sum_{z=1}^{K} \rho_{q,z} \cdot Expertise(u,z)) \cdot (\sum_{z=1}^{K} \rho_{q,z} \cdot Descriptive(u,z)) \\
&= \sum_{z=1}^{K} \rho_{q,z} \cdot UTCount(u,q) \cdot Expertise(u,z) \\
&\cdot \sum_{z=1}^{K} \rho_{q,z} \cdot Descriptive(u,z)
\end{aligned} \tag{4.11}$$

$$UTCount(u,q) = \left| T_u \bigcap T_q \right| \tag{4.12}$$

$$\rho_{q,z} \propto \sum_{w:\mathbf{w}_q} \varphi(z,w) + \sum_{t:\mathbf{t}_q} \psi(z,t) \tag{4.13}$$

where $\varphi(z,w)$ and $\psi(z,t)$ in Eq.4.13 can be estimated from UTAM; $\mathbf{w}_q$ and $\mathbf{t}_q$ are the word set and tag set of question q. We use $Expertise(u,q)$ to denote the expertise of user u for the question q, and $Descriptive(u,q)$ is the descriptive ability of user u for question q. Specifically, $Expertise(u,q)$ and $Descriptive(u,q)$ can be written as follows.

$$Expertise(u,q) = UTCount(u,q) \cdot \sum_{z=1}^{K} \rho_{q,z} \cdot Expertise(u,z)$$

$$Descriptive(u,q) = \sum_{z=1}^{K} \rho_{q,z} \cdot Descriptive(u,z)$$

Here $Descriptive(u,z)$ is the descriptive ability of user u under topic z, and $Expertise(u,z)$ denotes the expertise of user u under topic z. $UTCount(u,q)$ represents the count of tags from question q which are also appeared in the user u's previously answered questions. Here T_u denotes the tag set of user u, and T_q is the tag set of question q.

In our USTA method, the two abilities $Expertise(u,z)$ and $Descriptive(u,z)$ are expressed by the saliency scores $SS_z^e(u)$ (refer to Eq.4.5) and $SS_z^f(u)$ (refer to Eq.4.4), respectively. It can be seen from Eq.4.11 that the recommendation score of each candidate user is based on both descriptive ability and expertise. However, the important descriptive ability was treated as expertise in CQARank [110].

A user u with the highest ability for a question q is more likely to be highly ranked even if the topical similarity between user u and question q is not high. Therefore we do not consider it in user recommendation score calculation.

Table 4.3: Results on expert user recommendation experiment.

Methods	Spearman	Kendall	nDCG
USTA	0.2416	0.2010	0.9167
UTAM	0.2192	0.1881	0.9157
CQARank	0.2221	0.1851	0.9150
TEM	0.2006	0.1714	0.9135

The candidate user set $\mathcal{U}$ for a question q is composed of all the users who post answers to this question. Because some users can have more than one answer to a single question, the averaged vote scores of each user's answers to a single question are viewed as the final voting scores. Then we can get the ground truth user ranking list with respect to a question q.

To evaluate our method and baselines, the popular metric nDCG (normalized discounted cumulative gain) [41] is used to assess the ranking accuracy. The higher value signifies the better performance. In addition, the Kendall tau and Spearman's rho rank correlation coefficients are employed to measure the correlation between the rank lists of users via all methods and the ground truth ranking list.

The performance for expert recommendation is shown in Table 4.3. It can be seen from Table 4.3 that our UTAM model performs better than TEM model, and our USTA method is the best among all for all metrics. The two-tailed paired t-test under 5% significance level shows the improvements over baselines. It means that the rank list of users by our USTA method is closer to the real rank list of users. The superior performance of our USTA method demonstrates that the way of user ability modelling by our models can improve the performance of expert user recommendation.

4.4.4 Evaluation on Recommending Answers for a New Question

Given the question q and candidate answer set $\mathcal{A}$. It is interesting and necessary to recommend highly ranked answers to the question. The recommendation score $RS(a, q)$ associated with answer a and question q can be formulated as follows.

$$\begin{aligned} &RS(a, q) \\ &= sim(a, q) \cdot Expertise(u, q) \cdot Descriptive(u, q) \end{aligned} \tag{4.14}$$

where the definition of expertise $Expertise(u, q)$ and descriptive ability $Descriptive(u, q)$ have been given in Section 4.4.3. Here the topical similarity, $sim(a, q)$, between q and a is defined as the Cosine similarity between the answer topic distribution $\boldsymbol{\rho}_a$ and the question topic distribution

Table 4.4: Results on answers recommendation experiment.

Methods	Spearman	Kendall	nDCG
USTA	0.2551	0.2126	0.9153
UTAM	0.2172	0.1807	0.9128
CQARank	0.2188	0.1861	0.9125
TEM	0.1958	0.1660	0.9107

$\boldsymbol{\rho}_q$.

$$sim(a,q) = \frac{\boldsymbol{\rho}_a \cdot \boldsymbol{\rho}_q}{\|\boldsymbol{\rho}_a\| \, \|\boldsymbol{\rho}_q\|} = \frac{\sum_{z=1}^{K} \rho_{a,z} \cdot \rho_{q,z}}{\sqrt{\sum_{z=1}^{K} \rho_{a,z}^2} \cdot \sqrt{\sum_{z=1}^{K} \rho_{q,z}^2}} \tag{4.15}$$

where $\rho_{q,z}$ has been given in Eq.4.13. We define $\rho_{a,z}$ as: $\rho_{a,z} \propto \sum\limits_{w:\mathbf{w}_a} \varphi(z,w) + \sum\limits_{t:\mathbf{t}_a} \psi(z,t)$. Here $\mathbf{w}_a$ and $\mathbf{t}_a$ are the word set and tag set of answer a.

For each question q, the ground truth ranking list of answers is defined by ranking the vote scores of answers in the test set. For each method, the ranking list of answers to a given question is obtained by ordering the recommendation scores. For all methods, both the Q-A topical similarity and the ability of answerers are modeled, whereas the difference is the way of user ability modelling. We not only learn the expertise of users based on the voting scores of their answers, but also consider the users' descriptive ability via their question posts.

It is obvious from Table 4.4 that our UTAM model still shows superiority compared with TEM, and our USTA method outperforms all the baseline methods in terms of Kendall and Spearman correlation coefficients as well as nDCG. The two-tailed paired t-test under 5% significance level also shows the improvements over baselines. The comparison demonstrates that the answer ranking list obtained by our USTA method is closer to the ground truth list than the baseline methods, which implies that our method can better depict the ability of answerers.

4.4.5 Experiment on Similar Questions Retrieval

In CQA sites, not all the query questions can get direct answers from other users. Sometimes the askers may receive a link to other similar questions instead. As [110], we also select 1000 query questions from the dataset. Note that their similar questions are in the training dataset which are viewed as ground truth questions.

For the experimental study, the candidate question set is formed by the ground-truth similar questions together with another 1000 randomly selected questions from the training set.

We compute the topical similarity between the candidate questions and the query question in each method. After that we get a ranking list of all the candidate questions in terms of the recommendation score $RS(q_{query}, q_{candi})$.

In this task, if we only consider the topical similarity between the query question and candidate question, the performance of our method will be similar to the baseline methods. To explore the important tag information in questions, we also consider the effect of the tag similarity, we name it as "USTA+SimTag", then the recommendation scores can be defined as Eq.4.16.

$$\begin{aligned} RS(q_{query}, q_{candi}) &= (1 + Jaccard(T_{query}, T_{candi})) \\ &\cdot sim(\rho_{q_{query}}, \rho_{q_{candi}}) \end{aligned} \tag{4.16}$$

where $Jaccard(T_{query}, T_{candi})$ is the Jaccard Index for measuring the tag similarity between the tag set T_{query} of query question and T_{candi} of candidate question. $\rho_{q_{query}}$ and $\rho_{q_{candi}}$ are the topic distributions for query question and candidate similar question respectively, which can be calculated according to Eq.4.13. Here, $sim(\cdot, \cdot)$ still represents the Cosine similarity between two distributions.

We evaluate the performance of our USTA method and baselines based on the following metrics:

- Mean Averaged Precision (MAP): the mean of the averaged precision scores for each query question.

- The average rank of similar questions (avgR): the average rank of ground-truth similar questions among the candidate questions.

- Mean reciprocal rank (MRR): the average of the reciprocal ranks for query questions.

- Cumulative distribution of ranks (CDR): $CDR@x$ is the percentage of query questions whose similar questions are in the top x of the ranking list of candidate similar questions.

The higher the value of MAP, MRR and CDR, the better the performance is, while it is contrary for the average rank of similar questions.

Table 4.5 gives the result for recommending similar questions, we can observe from Table 4.5 that the performance achieves evident improvement when the tag information is considered for all metrics. Each topic are associated with many tags, and each tag corresponds to a subtopic. Therefore the more tags two questions share, the more similar they are. Note that there are very few ground truth questions among the large candidate questions for each query question. So the values of MAP are not high for all methods.

Table 4.5: Results on similar questions recommendation experiments.

Methods	MAP@20	MAP@50	avgR	MRR	CDR@20	CDR@50
USTA+SimTag (UTAM + SimTag)	0.0510	0.0247	25	0.3695	0.718	0.868
CQARank (TEM)	0.0362	0.0202	48	0.2048	0.507	0.719

4.4.6 Time Analysis

CQARank and UTAM are built based on TEM model and UTAM model, respectively. The averaged training time for each iteration of the Gibbs sampling procedure in TEM model is about 165.39 seconds, while it is around 188.45 seconds for the UTAM model. It can be seen that the time cost for them is close.

4.4.7 Discussion

Although the graphical notations of our UTAM model and the existing TEM model shown in Figure 4.1(a) and Figure 4.1(b) look similar, the difference is obvious: 1) The voting of them shows different meanings; 2) The topic distributions of questions and answers of each user are usually different. Due to the different properties of questions and answers, we learn them separately; 3) The voting of questions is based on question itself, whereas the voting of each answer is given based on the relevance and accuracy between the answer and its corresponding question.

The main comparison we focused on in this task is between the existing TEM and our UTAM, which are the basis of CQARank and our USTA method respectively.

4.5 Experimental Analysis on User-Tag Based Expert Recommendation

4.5.1 Experimental Setup

Stack Overflow, a large popular CQA service, has massive amount of questions and answers in the field of computer programming. A publicly available data dump from Stack Overflow ranging from August 2008 to August 2010 is used as the dataset in this task.

To prepare the training and testing datasets, we performed the following steps: 1) We collect questions Q_1 between August 1st 2009 and August 1st 2010 with each question having more than 15 answers. 2) Then we find out all the answers A_1 of these selected questions. 3) Get the user list UL_1 from the distinct users of these answers A_1. 4) We collect all the answers A_2 between August

1st 2008 and July 31st 2009, where the users of these answers should be in the user list UL_1. 5) In these collected answers A_2, the users with more than 50 answers will be considered, and their corresponding answers A_3 will be kept. 6) Find out the distinct user list UL_2 of these answers A_3. 7) Collect answers A_4 from A_1 whose users are in the user list UL_2. 8) All the answers from A_4 to the same questions are grouped together, we keep the groups with size > 5, then the corresponding answers A_5 are selected. 9) We collect all the answers in A_3 and their corresponding questions as the training set, and the answers in A_5 and their corresponding questions as the testing set. After the above steps, we can finally get 14986 tags and 1425 users.

4.5.2 Baseline Methods

We compare our user-tag PMF based method with the up-to-date TEM model [110], CQARank [110], UTAM model [107] and USTA [107]. In the following subsections, we will represent the experimental analysis and comparisons in expert ranking and recommendation for a new question.

In our user-tag PMF based method, we only use the voting scores of answers and the tags of the corresponding questions to learn the user expertise. In other words, we learn user expertise under tags. However, in the above mentioned baseline methods, besides the voting scores and tags, the whole body of answers are also used to model the user expertise. Specifically, as for TEM model, it depicts user expertise based on topics, where each topic can be represented by several tags and other words. Hence, our method can be viewed as expert modelling based on subtopics.

In our method, the number of latent features is set as 10, the learning rate is 0.1, and we let $\lambda_U = \lambda_Y = 0.01$, and the iteration is set as 500 times. For the baseline methods, we set the topic number K as 10, the number of expertise levels $E = 10$, and other settings are the same as those in [110]. The damping factor for saliency scores is set as $\lambda = 0.2$ in both CQARank and USTA methods.

The experiments for our method and baselines are conducted on a PC with Pentium Dual-core 2.3 GHz CPU and 4.0 GB RAM.

4.5.3 Evaluation on Expert User Recommendation

Given a new question, the objective of this task is to get a ranking list of users who are the potential answerers with high expertise. In the test set, the ground truth is the ranking list of users generated by sorting their votes of answers.

We use recommendation score $ReS(u, q)$ to depict the expertise for each user $u (u \in \mathcal{U})$ regarding a question q, then we can obtain a ranked user list according to the recommendation

Table 4.6: Results on expert user recommendation experiment.

Methods	nDCG	Pearson	Kendall
User-Tag PMF Based Method	0.8485	0.1012	0.0805
TEM	0.8201	-0.0441	-0.0354
UTAM	0.8279	-0.0058	-0.0032
CQARank	0.8346	0.0331	0.0281
USTA	0.8388	0.0594	0.0437

scores. The larger the score, the higher the rank is.

For our PMF based method, the recommendation score $ReS_{our}(u,q)$ is defined in Eq.4.17, where N_t denotes the number of tags in question q, which is based on the average expertise value of the user-tag expertise score $Ept(u,t)$ of user u and each tag t of question q.

$$ReS_{our}(u,q) = \frac{1}{N_t}\sum_{t=1}^{N_t} Ept(u,t) \tag{4.17}$$

As for the TEM model, $ReS_{TEM}(u,q)$ can be written in the form of Eq.4.18,

$$ReS_{TEM}(u,q) = Ept(u,q) = \sum_{z=1}^{K} \rho_{q,z} \cdot Ept(u,z) \tag{4.18}$$

where $Ept(u,z)$ represents the expertise of user u for topic z, and $\rho_{q,z} \propto \sum_{w:\mathbf{w}_q} \varphi(z,w) + \sum_{t:\mathbf{t}_q} \psi(z,t)$. Here $\varphi(z,w)$ and $\psi(z,t)$ denote the topic specific word distribution and topic specific tag distribution in TEM model, respectively. For a question q, we use $\mathbf{w}_q$ and $\mathbf{t}_q$ to represent its word set and tag set.

For a question q, the answerer set $\mathcal{U}$ consists of all the users who answered this question. In many cases, one can post multiple answers to a single question, hence we use the averaged voting scores of each user's answers to a question. The ground-truth user ranking list is obtained by sorting these scores.

There are three criteria used in the performance evaluation, namely, nDCG (normalized discounted cumulative gain) [41], the Kendall tau and pearson rank correlation coefficients, which are also adopted in [110] to measure the correlation between the ground-truth ranking list and the obtained user list. For the three criteria, the higher value means the better performance.

Table 4.6 shows the results for expert learning by different methods. It is obvious from Table 4.6 that our PMF based method performs better than TEM model and other baseline methods (i.e., CQARank, UTAM and USTA) especially in terms of Pearson and Kendall, which indicates that our method based on tags can better represent the user expertise than baseline models. We also conducted significance test for each method based on 5 runs, which demonstrates that the improvement is obvious.

4.5.4 Time Analysis

We also conducted running time analysis on our user-tag PMF based approach and the baseline methods, the averaged training time for each iteration in our user-tag PMF based method is about 1.03 seconds, while it is around 121.43 seconds for the TEM model and 181.65 seconds for the UTAM model. As for CQARank, the time is not only cost on the training of its base model (TEM model), but also cost on the link structure analysis. Similarly, the time cost on USTA is more than that on its base model (UTAM model). It is evident that our PMF based approach requires less time consumption than the baseline models. It indicates that our PMF based approach is more applicable in large scale CQA services.

4.6 Summary

In community question answering sites, the expert users are key resources, which can provide satisfactory answers to the askers and promote the well development of CQA services. The existing methods only consider the expertise of users, where the voting scores of questions were also viewed as the reflection of users expertise besides the voting of answers. In fact, the two types of votes represent different meanings. In this chapter, we propose to model the topical expertise based on answers and descriptive ability of users based on questions. Furthermore, the social link structures in CQA sites are introduced into the user ability modelling. Existing methods consider that the user expertise is defined on certain topics. Generally, such topics are too general, whereas question tags can be more informative and valuable than the topic of each question. To this end, we study the user expertise under tags rather than topics. The experiments conducted on the large CQA dataset from a very popular Stack Overflow service, show that our UTAM, USTA and user-tag PMF based models can perform competitively compared with the up-to-date methods in identifying experts, recommending suitable answers and retrieving similar questions. Furthermore the user-tag PMF based method is more competitive in terms of both effectiveness and efficiency.

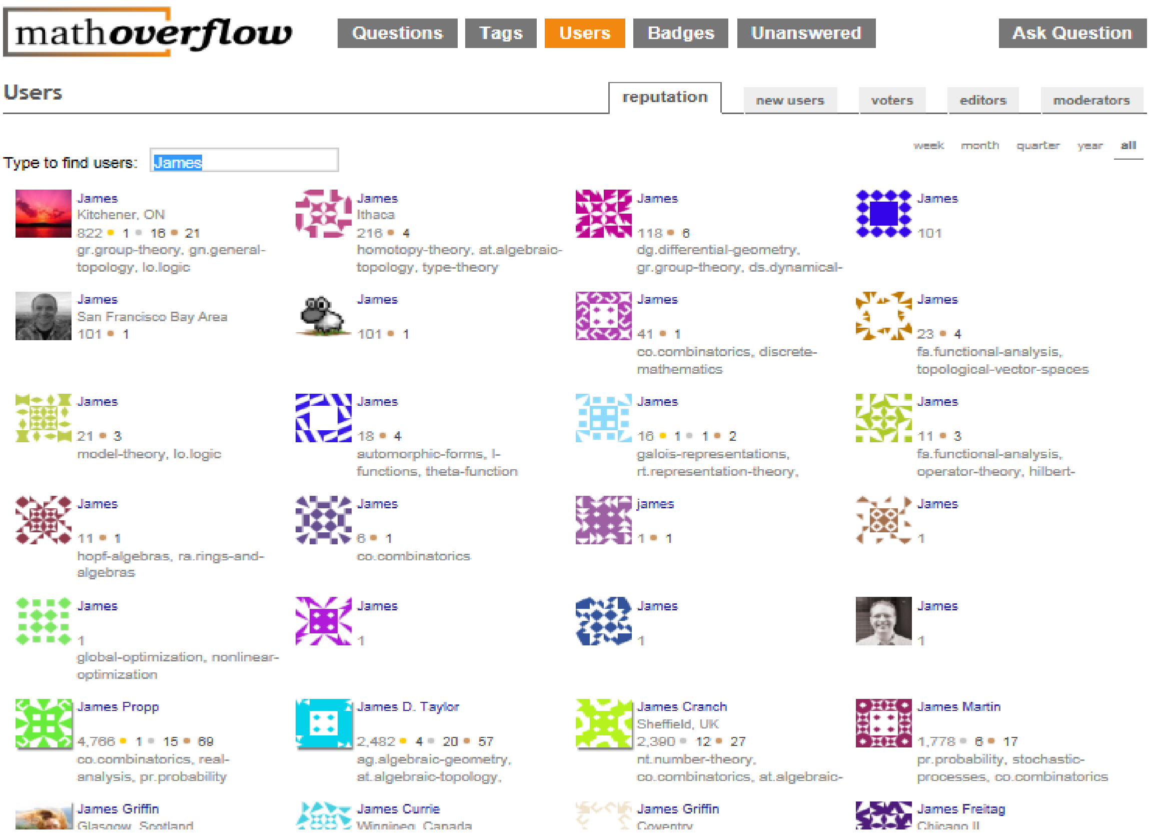
Chapter 5

User Name Disambiguation in Community Question Answering

Community question answering sites provide us convenient and interactive platforms for problem solving and knowledge sharing, which are attracting an increasing number of users. Accordingly, it will be very common that different people have the same user name. When a query question is given, some potential answer providers would be recommended to the asker in the form of user name. However, some user names are ambiguous and not unique in the community. As Figure 5.1, there are a number of users named "James" in MathOverflow[1], it will get difficult and time consuming for the asker to decide which homepage to visit.

In some cases, an off-line person asks people around a difficult question orally, then he/she may be recommended by word of mouth to visit the CQA homepages of some potential answer providers. However, the links to their homepages are not provided sometimes, then the asker has to search them according to the provided user names. Some user names are unique, and they can easily access the historical QA records of these potential answer providers. However, some are very common and ambiguous, accordingly, many users with the same user name will be displayed. As for user recommendation, when some user names are ambiguous, the askers will be thrown into another dilemma. To help question askers match the ambiguous user names with the right people, in this chapter, we propose to disambiguate same-name users by ranking their tag-based relevance to a query question. To our knowledge, this is the first work on user name disambiguation in community question answering. Although there have been some studies on user name disambiguation in bibliographic citation records [26, 36, 100], the related methods

[1] `http://mathoverflow.net/`



Users

Type to find users: James

James
Kitchener, ON
822 ● 1 ● 16 ● 21
gr.group-theory, gn.general-topology, lo.logic

James
Ithaca
216 ● 4
homotopy-theory, at.algebraic-topology, type-theory

James
118 ● 6
dg.differential-geometry, gr.group-theory, ds.dynamical-

James
101

James
San Francisco Bay Area
101 ● 1

James
101 ● 1

James
41 ● 1
co.combinatorics, discrete-mathematics

James
23 ● 4
fa.functional-analysis, topological-vector-spaces

James
21 ● 3
model-theory, lo.logic

James
18 ● 4
automorphic-forms, l-functions, theta-function

James
16 ● 1 ● 1 ● 2
galois-representations, rt.representation-theory,

James
11 ● 3
fa.functional-analysis, operator-theory, hilbert-

James
11 ● 1
hopf-algebras, ra.rings-and-algebras

James
6 ● 1
co.combinatorics

james
1 ● 1

James
1

James
1
global-optimization, nonlinear-optimization

James
1

James
1

James
1

James Propp
4,766 ● 1 ● 15 ● 69
co.combinatorics, real-analysis, pr.probability

James D. Taylor
2,482 ● 4 ● 20 ● 57
ag.algebraic-geometry, at.algebraic-topology,

James Cranch
Sheffield, UK
2,390 ● 12 ● 27
nt.number-theory, co.combinatorics, at.algebraic-

James Martin
1,778 ● 6 ● 17
pr.probability, stochastic-processes, co.combinatorics

James Griffin
Glasgow, Scotland

James Currie
Winnipeg, Canada

James Griffin
Coventry

James Freitag
Chicago IL

Figure 5.1: An example of users with given name "James" in MathOverflow community.

are not directly applicable to our work. In this chapter, to disambiguate the same-name users, we present a simple vector-style tag-based method, *relTagVec*, to learn the relevance between each user and the question by comparing their tag lists, where each tag is represented by a vector. Then the one who has the highest relevance score will be the right person to recommend. Experimental results on three CQA datasets from StackExchange[2] network demonstrate that our method is very effective for disambiguating user names in community question answering, and performs much better than the baseline methods.

Chapter outline

The remainder of this chapter is organized as follows: We describe the framework of our method in Section 5.1. Section 5.2 introduces the experimental setup. Section 5.3 shows the experimental results and analysis on real CQA datasets. Finally, we conclude this chapter in Section 5.4.

[2] http://stackexchange.com/

5.1 Framework of Our Method

In this section, the concrete steps of our *relTagVec* method are presented and explained.

5.1.1 Computing User Relevance to the Questions

For each user u, we can get a list of tags, T_u, from the questions to which he/she has recently answered. For each question q, the corresponding tag list can be represented as T_q. We use word2vec [68] technique to compute the vector representation of all the tags. And then our relevance value $relevance(u, q)$ of user u over q can be represented as follows.

$$\begin{aligned} & relevance(u, q) \\ & = \frac{1}{|T_q|} \sum_{i=1}^{|T_q|} \max_{j=1,2,\ldots,|T_u|} (sim(\mathbf{v}_i^{T_q}, \mathbf{v}_j^{T_u}) \cdot w_j^{T_u}), \end{aligned} \tag{5.1}$$

where $\mathbf{v}_i^{T_q}$ is the vector representation for the i-th tag in the tag list of question q. Accordingly, $\mathbf{v}_j^{T_u}$ is the vector for the j-th tag in the tag list of user u. Here $sim(\mathbf{v}_i^{T_q}, \mathbf{v}_j^{T_u})$ denotes the cosine similarity between $\mathbf{v}_i^{T_q}$ and $\mathbf{v}_j^{T_u}$. In addition, $w_j^{T_u}$ is the weight of j-th tag in the tag list of user u, which can be represented as $w_j^{T_u} = 1/(1 + \exp(-N_j^{T_u}))$. Here, $N_j^{T_u}$ is the number of times the j-th tag of user u appearing in the questions to which the user u has answered.

5.1.2 Selecting the User with Highest Relevance Value

When we get each relevance value $relevance(u, q)$ of candidate users to the query question q, the user with highest relevance value will be considered as the right person to recommend. Here we use $u_{predicted}^{q}(username)$ to denote the predicted user with the name "username" for recommendation over question q.

5.1.3 Recommending Ranked User List

In many cases, a considerable number of users share the same user name, then the prediction to the target person is getting difficult based on insufficient historical data, and the prediction accuracy will be low. It is very necessary to provide a ranking list to the asker.

For a query question q, we rank the candidate users to generate a ranking list based on relevance scores $relevance(u, q)$ in descending order. Then the askers just need to check the top-ranking users, which is time-saving.

Table 5.1: Statistics of all the datasets used for Type 1 in the experiments.

Dataset	Initial # Train/# Evaluation	# Ambiguous user names	# Total user names	# Proportion of ambiguous user names
Travel	13940/12685	502	12170	502/12170=4.12%
Cooking	27468/11260	740	16545	740/16545=4.47%
MathOverflow	48357/97444	2218	30574	2218/30574=7.25%

5.2 Experimental Setup

In this chapter, two types of user names are considered.

Type 1: Each provided ambiguous user name is exactly the *DisplayName* of the target user.

Type 2: The recommendation is only given in the form of each target user's first name. For example, a user named "Tom Smith" is mentioned in the name of "Tom" instead. However, there are many members named "Tom" in the community.

5.2.1 Datasets and Settings

In our experiments, three Data Dumps[3] from Travel[4], Seasoned Advice (Cooking)[5] and MathOverflow communities are used to evaluate our method. Note that all the user names are case insensitive in our experiments.

The statistics of all the datasets used in the experiments, including the train/evaluation split and the proportion of ambiguous user names, are shown in Table 5.1 and Table 5.2 for Type 1 and Type 2 respectively. In each of these two tables, the first column denotes the datasets for the experiments, the second column shows the initial train/evaluation split. The numbers of ambiguous user names and the total user names are listed in the third and fourth columns respectively. The last column gives the proportion of ambiguous user names.

Travel: We use a Travel Data Dump ranging from June 2011 to September 2014. First, the dataset is divided into two parts, the data before 2013-05-09 is viewed as historical data for training (13940 posts), while the remainder is used for evaluation (12685 posts).

For Type 1, firstly, from the historical set we select all the user names associated with at least two different users. Then the userIds of all the users who share the same user name will

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

[4] `http://travel.stackexchange.com/`

[5] `http://cooking.stackexchange.com/`

Table 5.2: Statistics of all the datasets used for Type 2 in the experiments.

Dataset	Initial # Train/# Evaluation	# Ambiguous user names	# Total user names	# Proportion of ambiguous user names
Travel	13940/12685	1691	13359	1691/13359=12.66%
Cooking	27468/11260	3150	18955	3150/18955=16.62%
MathOverflow	48357/97444	7880	36236	7880/36236=21.75%

be selected, and then we collect all their previous Q&A records (833 posts associated with 231 different users). Based on the userIds of these historical Q&A records, the questions answered by the corresponding users are selected from the initial evaluation dataset. Then we build the final evaluation data in the form of triples (question, user name, userId). Here the user name is ambiguous, and the user with this userId is a **gold standard** answer provider for this question. The final evaluation dataset contains 298 (question, user name, userId) records. For each ambiguous user name, the associated users with this name form the candidates. Note that each gold standard userId is known in evaluation set without manual annotation.

As for Type 2, we first select all the one-word user names from historical set, then all the user names containing these given names are selected. And then the userIds associated with these given names are collected from historical set, the remainder steps are similar to Type 1.

Cooking: The Seasoned Advice (Cooking) Data Dump is dated from July 2010 to September 2014. For Type 1, we preprocess it in the same way as that for Travel Data Dump. Here the initial historical set is composed of the data (27468 posts) before 2013-03-10, and the rest are used for evaluation (11260 posts). In historical set, we finally collect 3306 Q&A posts from 982 different users for training. And we get 284 (question, user name, userId) records for the evaluation set. The preprocessing for Type 2 is similar to that in Travel set.

MathOverflow: The Data Dump for MathOverflow ranging from September 2009 to September 2014 is also publicly available. Here the data before 2011-02-05 is formed as initial historical data (48357 posts). For Type 1, we finally collect 4090 posts for training and 2770 (question, user name, userId) records for evaluation. All the preprocessing steps for both types are the same as those for Travel Data Dump.

All the experiments are performed on a PC with Pentium Dual-core 2.3 GHz CPU and 4.0 GB RAM. For the tag vector representation, word2vec continuous bag of words (CBOW) model [68] is used, and the vectors are got based on the question tags from the whole dataset. We set the dimension of each vector as 50, and the training is executed for 10 iterations.

5.2.2 Baselines

- *Random*: A predictor generates random ranking of candidate answer providers for each question.

- *relTitle-Avg*: Given the title $Title_q$ of a query question q, the titles $\{Title_{q_i \in Q_u}\}_{i=1}^{|Q_u|}$ of the previously asked and answered questions Q_u from each candidate user u are collected, then we compute the Jaccard similarity coefficient between $Title_q$ and each $\{Title_{q_i \in Q_u}\}_{i=1}^{|Q_u|}$, and then the averaged similarity value is calculated, which is considered as the relevance score of user u to question q.

- *relTitle-Max*: Different from *relTitle-Avg*, in *relTitle-Max*, the maximum Jaccard similarity value is computed instead of the averaged similarity value.

- *relTag*: Given a question q, the relevance value of user u over q, can be represented as $(\sum_{c \in (T_u \bigcap T_q)} W_c) \cdot \frac{|T_u \bigcap T_q|}{|T_q|}$, where the tag similarity is computed, T_u is the tag list of user u and T_q is the tag list of question q. W_c denotes the number of times tag c appearing in the questions that u has answered.

We also tried to use $\frac{|T_u \bigcap T_q|}{|T_u \bigcup T_q|}$ to replace $\frac{|T_u \bigcap T_q|}{|T_q|}$ above, however, the performance is reduced. One reason is that $|T_u| \gg |T_q|$ in general.

5.2.3 Evaluation Metrics

We use accuracy as the metric for the most likely user prediction evaluation. The representation of accuracy is shown as follows.

$$Accuracy = \frac{N_{(u_{predicted} == u_{true})}}{N_{records}},$$

where $N_{records}$ denotes the number of (question, user name, userId) records in the evaluation set, and $N_{(u_{predicted} == u_{true})}$ is the number of records whose answer providers have been correctly matched. Here $u_{predicted}$ denotes the predicted userId, and u_{true} is the ground-truth userId of a user name for a record. The higher accuracy, the better performance is.

Because some user names are shared by many users, we also evaluate the predicted ranking of the ground-truth[6] user by our method and baselines in terms of the following metrics.

[6] The real ranking for ground-truth user should be 1.

- The average rank of ground-truth users (*avgR*): the average rank of ground-truth users among the candidate users for the query questions.
- Mean reciprocal rank (*MRR*): the average of the reciprocal ranks of ground-truth users for the query questions.
- Cumulative distribution of ranks (*CDR*): *CDR@m* is the percentage of query questions whose ground-truth answer providers are in the top m of the ranking list of candidate users.

The mathematical expressions for *avgR*, *MRR* and *CDR@m* are shown as follows.

$$AvgR = \frac{1}{|Q|}\sum_{q\in Q} r^{q}_{u_{true}}$$

$$MRR = \frac{1}{|Q|}\sum_{q\in Q} \frac{1}{r^{q}_{u_{true}}}$$

$$CDR@m = \frac{|\{q \in Q | r^{q}_{u_{true}} \leq m\}|}{|Q|}$$

Here, q is the query question from the question set Q. The expression $r^{q}_{u_{true}}$ denotes the rank of the ground-truth user u_{true} among the candidate users for question q.

The higher the values of *MRR* and *CDR*, the better the performance is, while it is contrary for *avgR*.

5.3 Results and Analysis

We compare our *relTagVec* method with the above four baseline methods on *Travel*, *MathOverflow* and *Cooking* datasets under Type 1 and Type 2 separately. For each type and each dataset, all the methods are run 10 times, then the averaged results are reported.

5.3.1 Performance under Type 1

In Type 1, the candidate users share the same names. Table 5.3(a) shows the results for all the methods on *MathOverflow* dataset, as for the most likely user prediction, *relTagVec* method performs best with promising accuracy value 0.8625, which is much more competitive than the baselines. For the performance on the ranking of ground-truth users, *relTagVec* is still superior to others in terms of avgR, MRR, CDR@2 and CDR@5. In addition, both *relTitle-Max* and *relTitle-Avg* methods perform better than *random* method. The *relTag* method performs better than the two title-based methods. And *relTitle-Max* method can yield more accurate results than *relTitle-Avg*.

Table 5.3: Performance under Type 1.

(a) MathOverflow

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.4536	1.8944	0.6892	0.8284	0.9883
relTitle-Avg	0.6472	1.6296	0.7931	0.8607	0.9894
relTitle-Max	0.6986	1.5790	0.8185	0.8592	0.9894
relTag	0.8053	1.4067	0.8800	0.9021	0.9927
relTagVec	0.8625	1.2747	0.9148	0.9296	0.9978

(b) Cooking

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.2226	4.7102	0.4179	0.3957	0.6360
relTitle-Avg	0.6360	1.7138	0.7824	0.8304	0.9859
relTitle-Max	0.8551	1.3887	0.9078	0.9152	0.9859
relTag	0.8975	1.3180	0.9340	0.9435	0.9859
relTagVec	0.9329	1.1166	0.9609	0.9753	0.9965

(c) Travel

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.5235	1.5336	0.7535	0.9564	1.0
relTitle-Avg	0.8993	1.1376	0.9435	0.9631	1.0
relTitle-Max	0.9262	1.1107	0.9569	0.9631	1.0
relTag	0.9295	1.1107	0.9580	0.9597	1.0
relTagVec	0.9698	1.0335	0.9843	0.9966	1.0

In Table 5.3(b), we can observe that *relTagVec* method still performs better than the baselines on *Cooking* dataset, and *random* method is the worst choice again. As for title-based methods, *relTitle-Max* is still superior to *relTitle-Avg* especially on accuracy.

As for the performance on *Travel* dataset shown in Table 5.3(c), it can be seen that *relTagVec* method still yields superior results in terms of all the metrics. By contrast, *random* is less competitive. Note that their CDR@5 values are all 1, which means that all the questions whose ground-truth answer providers are in the top 5 of the candidate list.

It is obvious from Table 5.3 that *relTagVec*, *relTag*, *relTitle-Max* and *relTitle-Avg* can effectively disambiguate the user names given the query question with regard to different evaluation metrics. By contrast, *relTagVec* performs best in Type 1.

5.3.2 Performance under Type 2

Different from Type 1, given a question, under Type 2, the querying user name only contains one word, which is usually viewed as the first name of a user. In such case, the candidate set is composed of all the users with the same first name. Accordingly, the user name will be more ambiguous with larger candidate set.

As can be seen from Table 5.4(a) that our *relTagVec* method still shows very promising performance, which outperforms the baseline methods in terms of all the listed evaluation metrics on MathOverflow dataset. For the baselines, *relTag* performs better, *random* method yields very low accuracy. As for the two title-based methods, *relTitle-Max* is still better than *relTitle-Avg*.

From Table 5.4(b) and Table 5.4(c), it tends to the similar conclusion that our *relTagVec* method performs better than the baselines on both *Cooking* and *Travel* datasets with acceptable performance.

Overall, *relTagVec* outperforms baseline methods under both types. Comparing Table 5.3 with Table 5.4 on each dataset, we can easily notice that the performance under Type 2 is reduced on each dataset with regard to nearly all the metrics, which is in accord with the fact that the user names (only given names) are more ambiguous. Moreover, the performance on Travel dataset is better than that on Cooking set in both types, which can be partly explained by Figure 1.2(a) and Figure 1.2(c), where the user names are less ambiguous in Travel community than Cooking Community, hence the performance is better on Travel dataset.

Error Analysis: We perform error analysis for *relTagVec* method and find that some candidate users share very similar values of $relevance(u, q)$, which can increase error rate and the difficulty in identifying target users.

Table 5.4: Performance under Type 2.

(a) MathOverflow

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.1646	9.4405	0.3408	0.3072	0.5505
relTitle-Avg	0.3648	4.8563	0.5509	0.5669	0.8084
relTitle-Max	0.4910	4.4630	0.6359	0.6504	0.8354
relTag	0.6633	2.9299	0.7681	0.7876	0.9141
relTagVec	0.6947	2.1003	0.7991	0.8250	0.9413

(b) Cooking

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.1731	8.0061	0.3375	0.2933	0.5030
relTitle-Avg	0.4562	3.6558	0.6147	0.6191	0.8228
relTitle-Max	0.6680	3.1181	0.7569	0.7719	0.8391
relTag	0.7413	2.9063	0.8037	0.8045	0.8595
relTagVec	0.7719	2.2546	0.8459	0.8717	0.9369

(c) Travel

Methods	User Predicting	User Ranking			
	Accuracy	avgR	MRR	CDR@2	CDR@5
random	0.3199	3.6919	0.5230	0.5446	0.7609
relTitle-Avg	0.6987	1.6355	0.8221	0.8956	0.9646
relTitle-Max	0.8476	1.4200	0.9046	0.9326	0.9697
relTag	0.8998	1.2694	0.9377	0.9554	0.9832
relTagVec	0.9217	1.1700	0.9535	0.9731	0.9899

5.4 Summary

The rapid growth of social question answering services comes with the contributions from the increasing number of registered members. Accordingly, the phenomenon about users with the same user names is getting more and more prevalent. If a user name is shared by many people in the community, once you input the user name, the system will display all the related users, in this case, it will get difficult to find out the target user. In this chapter, given a question, we focus on the user name disambiguation of potential answer providers in CQA. We utilize the tag information of both users and the query question to compute the relevance values. Then the user with highest relevance is viewed as the target user. We also recommend the possible ranked user list when there are a great number of candidates. In addition, the title-based methods are introduced in evaluation. Experimental analysis on three CQA datasets show that our *relTagVec* method is simple but very effective in user name disambiguation.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize this thesis, then the main contributions are presented. Moreover, we discuss a number of limitations of this thesis. Finally, we provide several suggestions and directions for the future work.

6.1 Thesis Summary

In this thesis, we model the user information in social communities and networks. We develop several social sentiment-topic models for identifying communities from online social networks. Moreover, we propose novel models for learning expert users in CQA, in addition, the best answer recommendation and similar question retrieval problems are also studied. Finally, to disambiguate same-name users in CQA, an effective method is proposed. The following is the summary of the work conducted in this thesis.

6.1.1 Social Sentiment-Topic Based Community Discovery

In social networks, the interactions among users are very frequent by sending emails, posting tweets, and sharing comments online, etc. Such networks usually include rich sentiment information. Most traditional community discovery methods only consider the social links among users, which ignore the valuable content information. Recent studies have focused on community detection by integrating both links and content. However, these methods are not available for identifying sentiment-topic based communities. In Chapter 3, we propose three novel community discovery models, STC, ASTC and ASTCx by integrating social links, user topics and sentiment information to identify communities with different sentiment-topic distributions. Experimental results on several real-world datasets demonstrate the effectiveness of our proposed models. Moreover, we conduct comparison among these three new models.

6.1.2 Expert User Recommendation in CQA

The second contribution of this thesis is to propose novel models for expert user learning in question answering communities. In CQA, to reduce the number of unanswered questions and the time for askers to wait, it is very necessary to identify relevant experts or best answers for these questions. Recent studies considered that user expertise is reflected by the voting scores of both answers and questions. However, voting scores of questions are not really related to user expertise. In Chapter 4, firstly we propose a new probabilistic model, UTAM, to depict users' expertise based on answers and their descriptive ability based on questions. To exploit social information in CQA, a new method, USTA, is proposed, where the link analysis is considered. Extensive experiments on the large Stack Overflow dataset demonstrate that our methods can achieve comparable or even better performance than the state-of-the-art model proposed in [110]. The above methods are based on modelling user topics, whereas question tags can be more informative and valuable than the topic of each question. To this end, we study the user expertise under tags. Experimental evaluation on a large data set from Stack Overflow shows that our user-tag PMF based method performs better than the latest topic-based methods.

6.1.3 User Name Disambiguation in CQA

It is quite common that different people can have the same user name in CQA. When a query question is given, some potential answer providers would be recommended to the asker in the form of user name. However, some user names are ambiguous and not unique in the community. To help question askers match the ambiguous user names with the right people, in Chapter 5, we propose to disambiguate same-name users by ranking their tag-based relevance to a query question. Then the user who has the largest relevance score is considered as the target person to recommend. Experimental studies on three community question answering datasets demonstrate that our method is effective for disambiguating user names in community question answering.

6.2 Limitations

We propose novel models and make several contributions to the research work in this thesis. However, there are still a number of limitations to be considered. We present these weaknesses and discussions as follows.

- In user sentiment-topic based community discovery, we utilize subjectivity lexicon as the sentiment prior knowledge, which is helpful for us to know the sentiment of some words in a document (an email or a tweet). However, in some documents the sentiments of users are not explicitly demonstrated using sentiment words. One case is that the users express their opinions via emoticons, such as :), :-), =), :(, :-(, :D, etc. Here each emoticon is viewed as an emotional indication, which is widely used in the social media platforms, like twitter. Another case is that the opinion of users is conveyed by the rating score for a product or service. In our work, the above situations are not considered. In social networks, two users with weak ties but strong topical similarities are possible from the same community. The traditional evaluation metric, modularity, is mainly defined based on the topology of community, which is not suitable for our STC, ASTC and ASTCx models. However, there is still no satisfying evaluation metric for this kind of community discovery. In addition, the numbers of communities and topics in each dataset are set manually in the experiments. Identifying the number of topics and communities automatically might be a better choice. Our ASTCx can separate topic words and sentiment words to avioid the mixed topic-sentiment words. However, it would be better if the matching between the topic words and sentiment words is considered.

- For the expert user learning in CQA, the running time cost in the model inference of our model will be high in the large-scale datasets. It is very necessary to propose other method to speed up the inference process. Additionally, the comment information in CQA is also very important. A user who gives acceptable comment to an answer is supposed to be able to answer the corresponding question, although he/she didn't answer this question directly. It might be a good try to view the comment as a special answer to a question. However, in our models, the valuable comments are not considered. In our user-tag PMF based expert user recommendation method, The tags of each question are usually not independent, some of them can appear simultaneously in many questions. It might be useful to explore the relationship between tags and incorporate it in the study of expert recommendation in CQA. Although tags of each question are very informative, some tags are not really very close to the content of the question. It is necessary to update the tags by replacing the less representative tags with the keywords extracted from the question content.

- For the user name disambiguation, the limitations are shown as follows. First, the candidate users' previous comments can be another reflection of users' interest. However, it is not utilized in our method. Second, there are other kinds of ambiguous types needing to be considered, like misspelling, which is also very common in the real life. Third, our method is simple but effective, it is interesting to try other ways to compute the relevance between a user and a question.

6.3 Future Work

As for the above mentioned limitations, we suggest a number of approaches and directions for the future research study.

- To address the issue in user sentiment-topic based community discovery, we propose the following strategies. First, we intend to incorporate the emoticons as another kind of sentiment prior knowledge. In addition, it is useful to transfer the rating scores as the sentiment polarities. Second, we plan to work out a new evaluation metric for community discovery by considering both structures and topical similarities. Third, as for the automatic detection of the numbers of communities and topics, the nonparametric Bayesian modelling might be a good choice. Fourth, to effectively match the topic words with the sentiment words, the aspect sentiment analysis is the future work. Another direction is to investigate the evolution of communities with the change of users' sentiment topics.
- To overcome the problem of expert user learning in CQA, several suggestions are shown as follows. First, to improve the speed of inference of our models, we intend to use synchronized inference methods. Second, due to the importance of user comments, it is expected to consider the user comments together with their answers to model user expertise. Third, it would be interesting to compute the semantic similarity between tags, which can overcome the cold-start problem when a new tag is appeared. Fourth, to replace the less useful tags, we consider to extract the keywords from the content of the corresponding question by using the existing keyword extraction methods, like RAKE (Rapid Automatic Keyword Extraction) [88], TermExtractor [91].
- To solve the problem of user name disambiguation in CQA, there are some points to be considered. For one thing, it might be useful to exploit the historical comment information together with the answers for learning user relevance scores. For another, we consider to propose other methods to improve the performance of user name disambiguation in CQA.

Appendix A

STC Model

A.1 Derivation for Posterior Conditional Distributions

According to the graphical representation shown in Fig. 3.1 and the generative process, we have:

$$\begin{aligned}P(\mathbf{c}) &= \int P(\mathbf{c}|\psi)P(\psi|\mu)d\psi \\ &= \frac{\Gamma(\sum_{m=1}^{M}\mu_m)}{\prod_{m=1}^{M}\Gamma(\mu_m)}\frac{\prod_{m=1}^{M}\Gamma(\mu_m + D_m)}{\Gamma(\sum_{m=1}^{M}\mu_m + D_m)}\end{aligned} \tag{A.1}$$

$$\begin{aligned}P(\mathbf{z}|\mathbf{c}) &= \int P(\mathbf{z}|\mathbf{c},\theta)P(\theta|\alpha)d\theta \\ &= \prod_{m=1}^{M}\frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\frac{\prod_{k=1}^{K}\Gamma(\alpha_k + n_{m,k})}{\Gamma(\sum_{k=1}^{K}\alpha_k + n_{m,k})}\end{aligned} \tag{A.2}$$

$$\begin{aligned}P(\mathbf{l}|\mathbf{c},\mathbf{z}) &= \int P(\mathbf{l}|\mathbf{c},\mathbf{z},\pi)P(\pi|\gamma)d\pi \\ &= \prod_{m=1}^{M}\prod_{k=1}^{K}\frac{\Gamma(\sum_{s=1}^{S}\gamma_s)}{\prod_{s=1}^{S}\Gamma(\gamma_s)}\frac{\prod_{s=1}^{S}\Gamma(\gamma_s + n_{m,k,s})}{\Gamma(\sum_{s=1}^{S}\gamma_s + n_{m,k,s})}\end{aligned} \tag{A.3}$$

$$\begin{aligned}P(\mathbf{u}|\mathbf{c}) &= \int P(\mathbf{u}|\mathbf{c},\lambda)P(\lambda|\delta)d\lambda \\ &= \prod_{m=1}^{M}\frac{\Gamma(\sum_{p=1}^{P}\delta_p)}{\prod_{p=1}^{P}\Gamma(\delta_p)}\frac{\prod_{p=1}^{P}\Gamma(\delta_p + g_{m,p})}{\Gamma(\sum_{p=1}^{P}\delta_p + g_{m,p})}\end{aligned} \tag{A.4}$$

$$\begin{aligned}&P(\mathbf{w}|\mathbf{z},\mathbf{l}) \\ &= \int P(\mathbf{w}|\mathbf{z},\mathbf{l},\phi)P(\phi|\beta)d\phi \\ &= \prod_{k=1}^{K}\prod_{s=1}^{S}\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\frac{\prod_{v=1}^{V}\Gamma(\beta_v + n_{k,s,v})}{\Gamma(\sum_{v=1}^{V}\beta_v + n_{k,s,v})}\end{aligned} \tag{A.5}$$

Then the conditional posterior probability of c_d, z_t, and l_t can be written as follows.

$$\begin{aligned}
&P(c_d = m|\mathbf{c}_{-d}, \mathbf{u}, \mathbf{z}, \mathbf{l}, \mathbf{w}) \\
&= P(c_d = m|\mathbf{c}_{-d}, \mathbf{u}, \mathbf{z}, \mathbf{l}) \\
&= \frac{P(\mathbf{c}, \mathbf{u}, \mathbf{z}, \mathbf{l})}{P(\mathbf{c}_{-d}, \mathbf{u}_{-d}, \mathbf{z}_{-d}, \mathbf{l}_{-d}, \mathbf{u}_d, \mathbf{z}_d, \mathbf{l}_d)} \\
&= \frac{P(\mathbf{c}, \mathbf{u}, \mathbf{z}, \mathbf{l})}{P(\mathbf{c}_{-d}, \mathbf{u}_{-d}, \mathbf{z}_{-d}, \mathbf{l}_{-d}) P(\mathbf{u}_d) P(\mathbf{z}_d, \mathbf{l}_d)} \\
&\propto \frac{P(\mathbf{c}, \mathbf{u}, \mathbf{z}, \mathbf{l})}{P(\mathbf{c}_{-d}, \mathbf{u}_{-d}, \mathbf{z}_{-d}, \mathbf{l}_{-d})} \\
&\propto \frac{P(\mathbf{c})}{P(\mathbf{c}_{-d})} \cdot \frac{P(\mathbf{z}|\mathbf{c})}{P(\mathbf{z}_{-d}|\mathbf{c}_{-d})} \cdot \frac{P(\mathbf{l}|\mathbf{z}, \mathbf{c})}{P(\mathbf{l}_{-d}|\mathbf{z}_{-d}, \mathbf{c}_{-d})} \cdot \frac{P(\mathbf{u}|\mathbf{c})}{P(\mathbf{u}_{-d}|\mathbf{c}_{-d})} \\
&\propto \frac{D_m^{-d} + \mu_m}{\sum_{j=1}^{M} \mu_j + D - 1} \cdot \frac{\prod_{k \in \mathbf{z}_d} \prod_{i=0}^{f_{d,k}-1} (\alpha_k + n_{m,k}^{-d} + i)}{\prod_{i=0}^{f_d - 1} (\sum_{k=1}^{K} \alpha_k + n_{m,k}^{-d} + i)} \\
&\cdot \prod_{k \in \mathbf{z}_d} \frac{\prod_{s \in \mathbf{l}_{d_{(k)}}} \prod_{i=0}^{f_{d,k,s}-1} (\gamma_s + n_{m,k,s}^{-d} + i)}{\prod_{i=0}^{f_{d,k}-1} (\sum_{s=1}^{S} \gamma_s + n_{m,k,s}^{-d} + i)} \\
&\cdot \frac{\prod_{p \in \mathbf{u}_d} (\delta_p + g_{m,p}^{-d})}{\prod_{i=0}^{e_d - 1} (\sum_{p=1}^{P} \delta_p + g_m^{-d} + i)}
\end{aligned} \tag{A.6}$$

We can get the posterior distribution of z_t and l_t based on the obtained community assignment c_d for document d.

$$\begin{aligned}
&P(z_t = k, l_t = s|\mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d) \\
&= \frac{P(\mathbf{z}, \mathbf{l}, c_d, \mathbf{w})}{P(\mathbf{z}_{-t}, \mathbf{l}_{-t}, c_d, \mathbf{w})} \\
&= \frac{P(\mathbf{z}, \mathbf{l}, \mathbf{w}|c_d)}{P(\mathbf{z}_{-t}, \mathbf{l}_{-t}, \mathbf{w}_{-t}|c_d) P(w_t|c_d)} \\
&\propto \frac{P(\mathbf{z}, \mathbf{l}, \mathbf{w}|c_d)}{P(\mathbf{z}_{-t}, \mathbf{l}_{-t}, \mathbf{w}_{-t}|c_d)} \\
&\propto \frac{P(\mathbf{z}|c_d)}{P(\mathbf{z}_{-t}|c_d)} \cdot \frac{P(\mathbf{l}|\mathbf{z}, c_d)}{P(\mathbf{l}_{-t}|\mathbf{z}_{-t}, c_d)} \cdot \frac{P(\mathbf{w}|\mathbf{z}, \mathbf{l})}{P(\mathbf{w}_{-t}|\mathbf{z}_{-t}, \mathbf{l}_{-t})} \\
&\propto \frac{n_{c_d,k}^{-t} + \alpha_k}{\sum_{k=1}^{K} n_{c_d,k}^{-t} + \alpha_k} \cdot \frac{n_{c_d,k,s}^{-t} + \gamma_s}{\sum_{s=1}^{S} n_{c_d,k,s}^{-t} + \gamma_s} \cdot \frac{n_{k,s,v}^{-t} + \beta_v}{\sum_{v=1}^{V} n_{k,s,v}^{-t} + \beta_v}
\end{aligned} \tag{A.7}$$

References

[1] Y. Ahn, J. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[2] A. Arenas, J. Duch, A. Fernández, and S. Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9(6):176, 2007.

[3] S. Asur, S. Parthasarathy, and D. Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 913–921. ACM, 2007.

[4] H. Avron and L. Horesh. Community detection using time-dependent personalized pagerank. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1795–1803, 2015.

[5] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. In *International Conference on Applied Computing (IADIS 2005)*, pages 97–104, 2005.

[6] J. Baumes, M. Goldberg, and M. Magdon-Ismail. Efficient identification of overlapping communities. In *Proceedings of the 2005 IEEE international conference on Intelligence and Security Informatics*, pages 27–36. Springer-Verlag, 2005.

[7] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199. ACM, 2000.

[8] J. Bian, Y. Liu, D. Zhou, E. Agichtein, and H. Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th international conference on World wide web*, pages 51–60. ACM, 2009.

[9] M. Bouguessa, B. Dumoulin, and S. Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 866–874. ACM, 2008.

[10] L. Cai, G. Zhou, K. Liu, and J. Zhao. Large-scale question classification in cqa by leveraging wikipedia semantic knowledge. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1321–1330. ACM, 2011.

[11] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community qa. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, volume 11, pages 273–281, 2011.

[12] X. Cao, G. Cong, B. Cui, and C. S. Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of the 19th international conference on World wide web*, pages 201–210. ACM, 2010.

[13] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560. ACM, 2006.

[14] W. Chan, X. Zhou, W. Wang, and T.-S. Chua. Community answer summarization for multi-sentence question with group l 1 regularization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 582–591. Association for Computational Linguistics, 2012.

[15] B.-C. Chen, J. Guo, B. Tseng, and J. Yang. User reputation in a comment rating environment. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 159–167. ACM, 2011.

[16] Y. Cheng, Z. Chen, J. Wang, A. Agrawal, and A. Choudhary. Bootstrapping active name disambiguation with crowdsourcing. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1213–1216. ACM, 2013.

[17] Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. ACM, 2007.

[18] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72(2):026132, 2005.

[19] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[20] S. Dongen. A cluster algorithm for graphs. *Methods*, (Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, May 2000):1–40, 2000.

[21] T. Evans. Clique graphs and overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment*, 2010:P12037, 2010.

[22] T. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):016105, 2009.

[23] T. Evans and R. Lambiotte. Line graphs of weighted networks for overlapping communities. *The European Physical Journal B-Condensed Matter and Complex Systems*, 77(2):265–272, 2010.

[24] I. Farkas, D. Ábel, G. Palla, and T. Vicsek. Weighted network modules. *New Journal of Physics*, 9:180, 2007.

[25] D. Fenn, M. Porter, M. McDonald, S. Williams, N. Johnson, and N. Jones. Dynamic communities in multichannel data: An application to the foreign exchange market during the 2007-2008 credit crisis. *Chaos: An interdisciplinary journal of nonlinear science*, 19(3):033119, 2009.

[26] A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. Laender. Effective self-training author name disambiguation in scholarly digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 39–48. ACM, 2010.

[27] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[28] M. Girvan and M. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[29] S. Gregory. An algorithm to find overlapping community structure in networks. *Knowledge Discovery in Databases: PKDD 2007*, pages 91–102, 2007.

[30] S. Gregory. A fast algorithm to find overlapping communities in networks. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases-Part I*, pages 408–423. Springer-Verlag, 2008.

[31] S. Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12:103018, 2010.

[32] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:1, 2004.

[33] R. Guimerà, M. Sales-Pardo, and L. Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(3):036102, 2007.

[34] J. Guo, S. Xu, S. Bao, and Y. Yu. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 921–930. ACM, 2008.

[35] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–305. ACM, 2004.

[36] H. Han, H. Zha, et al. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'05)*, pages 334–343, 2005.

[37] T. Hao and E. Agichtein. Finding similar questions in collaborative question answering archives: toward bootstrapping-based equivalent pattern learning. *Information retrieval*, 15(3-4):332–353, 2012.

[38] F. Havemann, M. Heinz, A. Struck, and J. Gläser. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *Journal of Statistical Mechanics: Theory and Experiment*, 2011:P01023, 2011.

[39] P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social networks*, 5(2):109–137, 1983.

[40] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Tracking evolving communities in large linked networks. *PNAS*, 101(1):5249–5253, 2004.

[41] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[42] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 84–90. ACM, 2005.

[43] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 228–235. ACM, 2006.

[44] Z. Ji and B. Wang. Learning to rank for question routing in community question answering. In *Proceedings of the 22nd ACM international conference on information and knowledge management*, pages 2363–2368. ACM, 2013.

[45] Z. Ji, F. Xu, B. Wang, and B. He. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2471–2474. ACM, 2012.

[46] D. Jin, B. Yang, C. Baquero, D. Liu, D. He, and J. Liu. A markov random walk under constraint for discovering overlapping communities in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011:P05031, 2011.

[47] P. Jurczyk and E. Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 919–922. ACM, 2007.

[48] B. Karrer and M. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[49] S. Kelley. *The existence and discovery of overlapping communities in large-scale networks*. PhD thesis, Rensselaer Polytechnic Institute, 2009.

[50] M. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proceedings of the VLDB Endowment*, 2(1):622–633, 2009.

[51] Y. Kim and H. Jeong. Map equation for link communities. *Physical Review E*, 84(2):026110, 2011.

[52] K. Kloster and D. F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1386–1395. ACM, 2014.

[53] J. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2):026109, 2008.

[54] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11:033015, 2009.

[55] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

[56] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block models. *Technical report*, 2009.

[57] C. Lee, F. Reid, A. McDaid, and N. Hurley. Detecting highly overlapping community structure by greedy clique expansion. In *SNAKDD 2010*, pages 33–42. ACM, 2010.

[58] E. Leicht and M. Newman. Community structure in directed networks. *Physical Review Letters*, 100(11):118703, 2008.

[59] B. Li, Y. Liu, A. Ram, E. V. Garcia, and E. Agichtein. Exploring question subjectivity prediction in community qa. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 735–736. ACM, 2008.

[60] F. Li, M. Huang, and X. Zhu. Sentiment analysis with global topics and local dependency. In *Proceedings of AAAI*, pages 1371–1376, 2010.

[61] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[62] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 685–694. ACM, 2008.

[63] J. Liu, Y.-I. Song, and C.-Y. Lin. Competition-based user expertise score estimation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 425–434. ACM, 2011.

[64] X. Liu, W. B. Croft, and M. Koll. Finding experts in community-based question-answering services. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 315–316. ACM, 2005.

[65] G. S. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 33–40. Association for Computational Linguistics, 2003.

[66] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, 2007.

[67] A. McDaid and N. Hurley. Detecting highly overlapping communities with model-based overlapping seed expansion. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 112–119. IEEE Computer Society, 2010.

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

[69] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. Mccallum. Optimizing semantic coherence in topic models. In *Conference on Empirical Methods in Natural Language Processing*, pages 262–272, 2011.

[70] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107, 2008.

[71] M. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[72] M. Newman. Random graphs with clustering. *Physical review letters*, 103(5):58701, 2009.

[73] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[74] M. Newman and E. Leicht. Mixture models and exploratory analysis in networks. In *Proceedings of the National Academy of Science*, volume 104, pages 9564–9569, 2007.

[75] A. Padrol-Sureda, G. Perarnau-Llobet, J. Pfeifle, and V. Muntés-Mulero. Overlapping community search for social networks. In *Proceedings of the 26th International Conference on Data Engineering (ICDE)*, pages 992–995. IEEE, 2010.

[76] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking : Bringing order to the web. *Technical Report*, (1999-66), 1999.

[77] A. Pal, F. M. Harper, and J. A. Konstan. Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Transactions on Information Systems (TOIS)*, 30(2):10, 2012.

[78] A. Pal and J. A. Konstan. Expert identification in community question answering: exploring question selection bias. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1505–1508. ACM, 2010.

[79] G. Palla, A. Barabasi, and T. Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[80] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[81] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson. Social topic models for community extraction. In *The 2nd SNA-KDD Workshop*, volume 8, 2008.

[82] X. Qiu, L. Tian, and X. Huang. Latent semantic tensor indexing for community-based question answering. In *51th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 434–439, 2013.

[83] M. Qu, G. Qiu, X. He, C. Zhang, H. Wu, J. Bu, and C. Chen. Probabilistic question recommendation for question answering communities. In *Proceedings of the 18th international conference on World wide web*, pages 1229–1230. ACM, 2009.

[84] U. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[85] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[86] B. Rees and K. Gallagher. Overlapping community detection by collective friendship group inference. In *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 375–379. IEEE Computer Society, 2010.

[87] W. Ren, G. Yan, X. Liao, and L. Xiao. Simple probabilistic algorithm for detecting community structure. *Physical Review E*, 79(3):036111, 2009.

[88] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[89] M. Sachan, D. Contractor, T. Faruquie, and L. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 331–340. ACM, 2012.

[90] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

[91] F. Sclano and P. Velardi. Termextractor: a web application to learn the shared terminology of emergent web communities. *Enterprise Interoperability II: New Challenges and Approaches*, pages 287–290, 2007.

[92] H. Shen, X. Cheng, K. Cai, and M. Hu. Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, 388(8):1706–1712, 2009.

[93] A. Singh and K. Visweswariah. Cqc: classifying questions in cqa websites. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2033–2036. ACM, 2011.

[94] Y. Song, J. Huang, I. G. Councill, J. Li, and C. L. Giles. Efficient topic-based unsupervised name disambiguation. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 342–351. ACM, 2007.

[95] J. Sun, C. Faloutsos, S. Papadimitriou, and P. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 687–696. ACM, 2007.

[96] J. Sung, J.-G. Lee, and U. Lee. Booming up the long tails: Discovering potentially contributive users in community-based question answering services. In *Seventh International AAAI Conference on Weblogs and Social Media*, pages 602–610, 2013.

[97] L. Tang, H. Liu, and J. Zhang. Identifying evolving groups in dynamic multimode networks. *IEEE Transactions on Knowledge and Data Engineering*, 24(1):72–85, 2012.

[98] Q. Tian, P. Zhang, and B. Li. Towards predicting the best answers in community-based question-answering services. In *Seventh International AAAI Conference on Weblogs and Social Media*, pages 725–728, 2013.

[99] M. Tomasoni and M. Huang. Metadata-aware measures for answer summarization in community question answering. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 760–769. Association for Computational Linguistics, 2010.

[100] P. Treeratpituk and C. L. Giles. Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 39–48. ACM, 2009.

[101] J. Tyler, D. Wilkinson, and B. Huberman. Email as spectroscopy: automated discovery of community structure within organizations. In *Communities and technologies*, pages 81–96. Kluwer, BV, 2003.

[102] X. Wang, J. Tang, H. Cheng, and P. S. Yu. Adana: Active name disambiguation. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pages 794–803. IEEE Computer Society, 2011.

[103] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP'05*, pages 347–354, 2005.

[104] H. Wu, Y. Wang, and X. Cheng. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 99–106. ACM, 2008.

[105] J. Xie and B. Szymanski. Towards linear time overlapping community detection in social networks. In *Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 25–36, 2012.

[106] J. Xie, B. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Proceedings of the 11th IEEE International Conference on Data Mining Workshops*, pages 344–349. IEEE, 2011.

[107] B. Yang and S. Manandhar. Exploring user expertise and descriptive ability in community question answering. In *Proceedings of 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 320–327. IEEE, 2014.

[108] B. Yang and S. Manandhar. Stc: A joint sentiment-topic model for community identification. In *PAKDD Workshops: Trends and Applications in Knowledge Discovery and Data Mining*, pages 535–548. Springer, 2014.

[109] K.-H. Yang, H.-T. Peng, J.-Y. Jiang, H.-M. Lee, and J.-M. Ho. Author name disambiguation for citations using topic and web correlation. In *Research and advanced technology for digital libraries*, pages 185–196. Springer, 2008.

[110] L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen. Cqarank: jointly model topics and expertise in community question answering. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 99–108. ACM, 2013.

[111] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.

[112] S. Yoon, A. Jatowt, and K. Tanaka. Detecting intent of web queries using questions and answers in cqa corpus. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 352–355. IEEE Computer Society, 2011.

[113] S. Yoon, A. Jatowt, and K. Tanaka. Search intent discovery by structurization of community qa contents. In *Web Information Systems Engineering-WISE 2012*, pages 712–718. Springer, 2012.

[114] B. Zhang, T. K. Saha, and M. A. Hasan. Name disambiguation from link data in a collaboration graph. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 81–84. IEEE, 2014.

[115] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM, 2007.

[116] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 371–380. ACM, 2014.

[117] S. Zhang, R. Wang, and X. Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490, 2007.

[118] Z. Zhao, F. Wei, M. Zhou, and W. S. H. Ng. Cold-start expert finding in community question answering via graph regularization. In *International Conference on Database Systems for Advanced Applications*, 2015.

[119] D. Zhou, E. Manavoglu, J. Li, C. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th international conference on World Wide Web*, pages 173–182. ACM, 2006.

[120] G. Zhou, S. Lai, K. Liu, and J. Zhao. Topic-sensitive probabilistic model for expert finding in question answer communities. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1662–1666. ACM, 2012.

[121] G. Zhou, F. Liu, Y. Liu, S. He, and J. Zhao. Statistical machine translation improves question retrieval in community question answering via matrix factorization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 852–861, 2013.

[122] G. Zhou, K. Liu, and J. Zhao. Joint relevance and answer quality learning for question routing in community qa. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1492–1496. ACM, 2012.

[123] G. Zhou, Y. Liu, F. Liu, D. Zeng, and J. Zhao. Improving question retrieval in community question answering using world knowledge. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2239–2245. AAAI Press, 2013.

[124] T. C. Zhou, X. Si, E. Y. Chang, I. King, and M. R. Lyu. A data-driven approach to question subjectivity identification in community question answering. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[125] W. Zhou, H. Jin, and Y. Liu. Community discovery and profiling with social messages. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 388–396. ACM, 2012.

[126] Z.-M. Zhou, M. Lan, Z.-Y. Niu, and Y. Lu. Exploiting user profile information for answer ranking in cqa. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 767–774. ACM, 2012.