Ranking user generated content using topic models

Ma, Zongyang

2015

Ma, Z. (2015). Ranking user generated content using topic models. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65539>

<https://doi.org/10.32657/10356/65539>

NANYANG
TECHNOLOGICAL
UNIVERSITY

**RANKING USER GENERATED CONTENT
USING TOPIC MODELS**

MA ZONGYANG
SCHOOL OF COMPUTER ENGINEERING
2015

Ranking User Generated Content Using Topic Models

MA ZONGYANG

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2015

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Associate Professor Sun Aixin, for his patient guidance and advice. It is my pleasure to be his student. He taught me not only how to learn knowledge of information retrieval and data mining, but also how to do research. I also would like to thank Associate Professor Cong Gao, for his guidance and suggestions on my research.

I would like to express my deep gratitude to my friends for their friendship and selfless help. They include Yuan Quan, Li Chenliang, Zheng Xin, Cao Xin, Xu Zhiqiang, Chu Shumo, Chen Ning, Lin Wenqing, Zhu Diwen, Zhang Boshen, Zhang Jun, Tang Youze, Wang Sibao, Shi Yancheng, Zhao Peilin, Li Shaohua, Liu Yong, Zhang Huanhuan and Xu Chang.

Last but not least, I am much obliged to my parents, for their selfless love and full support.

Abstract

With the popularity of Web 2.0, more and more users express and share opinions through various online platforms. Example platforms include news websites that support user commenting like Yahoo! News, social network sites that allow users to post messages like Facebook and Twitter, and community-based question answering sites which let users ask and answer questions. As the result, a huge amount of User Generated Content (UGC) is accumulated online in the forms of comments, tweets, question and answer posts, and others.

Depending on the platform within which UGC is created, UGC may be associated with different types of attributes such as creator, time, location, text and social connections of its creator. On the other hand, UGC data from different platforms shares similar characteristics: huge amount, free writing style, and heterogeneous nature. More importantly, UGC data often demonstrates *master-slave* relationship. A comment is associated with a news article; a hashtag is an annotation of its embedded tweet; an answer does not exist without a question. Here, news articles, tweets, and questions are master documents while comments, hashtags, and answers are slave documents. Although topic modeling (*e.g.*, LDA and PLSA) has been widely used to model text collections, discovering fine-grained topics from UGC with the consideration of master-slave relationship remains an open and challenging problem. In this research, the generative process of UGC data is simulated using topic models for the ranking of slave documents of given master documents with the aim of reducing information overload. Depending on the platform that UGC data is created in, three

sub-problems are defined and addressed: (i) comment ranking for news articles, (ii) hashtag ranking for tweets, and (iii) answer ranking for questions.

Comment ranking is essential for identifying the important comments as a summary of user discussion for a news article. In this task, we assume that topics of slave documents cover the topics of their corresponding master document, and also the topics discussed solely in comments. For this problem, we propose two LDA-style topic models, namely, Master-Slave Topic Model (MSTM) and Extended Master-Slave Topic Model (EXTM). MSTM model constrains that the topics discussed in comments have to be derived from the commenting news article. EXTM model allows generating words of comments using both the topics derived from the commenting news article, and the topics derived from all comments themselves. Evaluated on Yahoo! News, the proposed models outperform baseline methods.

Hashtag ranking is important for tweet annotation and retrieval. Here, we assume that the topics of slave documents are the topical summary of their corresponding master documents. For this problem, we propose two PLSA-style topic models to model the hashtag annotation behavior. Content-Pivoted Model (CPM) assumes that tweet content guides the generation of hashtags, while Hashtag-Pivoted Model (HPM) assumes that hashtags guide the generation of tweet content. The experimental results demonstrate that CPM is most effective for ranking the most relevant hashtags of tweets.

Answer ranking enables users to easily pick up the best answers for questions. In this task, we assume that topics of slave documents and topics of their corresponding master documents are similar but words of slave topics and master topics are drawn from different vocabularies. For this problem, we propose a PLSA-style topic model, namely, Tri-Role Topic Model (TRTM), to model the tri-roles of users (*i.e.*, as askers, answerers, and voters, respectively) and the activities of each role including composing question, selecting question to answer, contributing and voting answers. Evaluated on Stack Overflow data, TRTM outperforms state-of-the-art methods for ranking high-quality answers for given questions.

These three problems are all on ranking UGC data from different platforms using topic models and the proposed topic models are extended depending on the master-slave structure of UGC data. For the problem of comment ranking, the slave documents (comments) are much shorter than their corresponding master document (news article). Our main concern is discovering topics from comments which reflect the topics of their news article as well as keeping topics merely discussed among comments. For the problem of hashtag ranking, the slave documents (hashtags) are extremely short, and sometimes the hashtag is just the abbreviation of one or a few words. Compared with comment ranking, hashtag ranking is more difficult and we thus introduce more factors (*e.g.*, user and time) to enrich the hashtag representation. Lastly, for the problem of answer ranking, the answer has an important feature of vote. It is challenging for us to model the voting behavior of users in a generative model. To address this task, we focus more on modeling the relationships between questions, answers, askers and answerers using the exponential KL-divergence function.

In this research, we define three ranking problems of User Generated Content. To address these problems, we propose several extended topic models to fit the characteristics and the structure of UGC data from different platforms. From Yahoo! News to Twitter, then to Stack Overflow, the features of the adopted data in our research are more and more complicated. The designed topic models include more features and relationships to more accurately simulate the generation process of UGC data. Experimental results show that our methods outperform baseline methods for all three problems.

Contents

Acknowledgments	i
Summary	iii
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 User Generated Content	1
1.2 Topic Discovery and Slave Document Ranking	3
1.3 Approaches and Methodologies	6
1.4 Contributions and Thesis Organization	11
2 Literature Review	14
2.1 Probabilistic Latent Semantic Analysis	14
2.1.1 Applications of PLSA	17
2.1.2 Extended PLSA Related to Our Work	22
2.2 Latent Dirichlet Allocation	24
2.2.1 Relaxing Assumptions of LDA	27
2.2.2 Applications of LDA	29
2.2.3 Extended LDA Related to Our Work	34
3 Comment Ranking for News Articles	38
3.1 Related Work	39
3.2 Comment Ranking and Selection	42

3.2.1	Master-Slave Topic Model (MSTM)	45
3.2.2	Extended Master-Slave Topic Model (EXTM)	48
3.2.3	Complexity Analysis	50
3.2.4	Salient Comment Selection	50
3.3	Experiments	52
3.3.1	Data Set	53
3.3.2	Evaluation	54
3.3.3	Topic Discovery	57
3.4	User Study	59
3.5	Summary	63
4	Hashtag Ranking for Tweet Annotation	64
4.1	Related Work	65
4.2	Hashtag Annotation Models	67
4.2.1	Content-Pivoted Model (CPM)	69
4.2.2	Hashtag-Pivoted Model (HPM)	70
4.2.3	Inference Algorithms	72
4.2.4	Complexity Analysis	73
4.2.5	Social Network Regularization	74
4.3	Experiments	75
4.3.1	Data Set	75
4.3.2	Evaluation by Perplexity	77
4.3.3	Topic Discovery	81
4.4	Applications	85
4.4.1	Hashtag Ranking for Tweet Annotation	85
4.4.2	Related Hashtags Discovery	90
4.5	Summary	93
5	Answer Ranking for Questions	94
5.1	Related Work	95

5.2	Tri-Role Topic Model	96
5.2.1	Notations	96
5.2.2	Model Description	97
5.2.3	The Inference Algorithm of TRTM	101
5.2.4	Complexity Analysis	102
5.3	Experiments	103
5.3.1	Topic Discovery	103
5.3.2	Answer Ranking for Questions	105
5.4	Summary	108
6	Conclusion and Future Work	109
6.1	Conclusion	109
6.2	Future Work	112
A	List of Publications	114
	Appendix - List of Publications	114
	References	116

List of Figures

2.1	Probabilistic Latent Semantic Analysis	15
2.2	Topic Model of W^4 (short for Who, Where, When and What)	18
2.3	Dual Role Model	21
2.4	Latent Dirichlet Allocation	24
2.5	correspondence Latent Dirichlet Allocation (corrLDA)	35
3.1	Topic-driven reader comment summarization	42
3.2	MSTM and EXTM. The dotted line illustrates slave topics are derived from news articles; the bold line illustrates the extended topics are derived from comments.	46
3.3	Number of comments per news article	53
3.4	Comment length distribution	53
3.5	Comment length percentile	54
3.6	Perplexity of MSTM and EXTM	56
3.7	User Evaluation	61
4.1	Graphical model representations of CPM (without dotted line) and CPM^{sn} (with dotted line).	69
4.2	Graphical model representations of HPM (without dotted line) and HPM^{sn} (with dotted line).	71
4.3	Hashtag frequency distribution and number of hashtags per tweet . .	76
4.4	Perplexity of CPM and HPM with varying α , β or K	78

4.5	The impact of social network regularization on <i>CPM</i> and <i>HPM</i> ($K=100$)	79
4.6	Top-10 topics to each hashtag ranked by probability $p(z h)$ in descending order	83
4.7	Hit rate of the four methods for top-5/top-10 hashtags	88
5.1	The tri-roles of a user and the activities	97
5.2	Tri-Role Topic Model	99
5.3	Topic distributions of asker role and answerer role from the same example user	105

List of Tables

3.1	Notations for MSTM and EXTM	45
3.2	Example topical keywords for master topics and their corresponding slave topics and extended topics. The master, slave, and extended topics are denoted by -M, -S, and -E respectively.	58
3.3	Least relevant extended topics derived from comments	59
4.1	Notations for CPM and HPM	67
4.2	Statistics of the data set	77
4.3	Example topics with <i>CPM</i> topical words, <i>CPM</i> topical hashtags and <i>HPM</i> topical words	82
4.4	Example topics found by <i>HPM</i> but not <i>CPM</i>	83
4.5	Topical keywords of the top-3 topics for each hashtag	84
4.6	Kappa scores between three pairs of volunteers (<i>v</i> 's)	92
4.7	Precision for related hashtag discovery with the best result in boldface	92
4.8	Top-5 most related hashtags to #prayforjapan and #movies , discovered by the four methods	93
5.1	Notations for TRTM	96
5.2	Example topics by TRTM from Stack Overflow	104
5.3	Topics of the same example user	105
5.4	nDCG of the three models; the best result for each topic number setting ($ Z =10, 20, 40$) is in boldface.	107

Chapter 1

Introduction

1.1 User Generated Content

With the various Web 2.0 platforms and applications, users nowadays are willing to express and share their opinions online. The Web 2.0 platforms also serve as communication channels for free communicating among users. Moreover, many traditional media are adding components to support social communications among their users. For example, Yahoo! News allows users to comment on their news articles. Social platforms like Twitter and Facebook, and community-based question answering (CQA) systems like Stack Overflow accumulate a huge amount of User Generated Content (UGC) in the forms of comments, tweets, question and answer posts. The User Generated Content is helpful for users to send and receive information in an efficient manner and in free style writing. Twitter user freely comments on ongoing events, posts personal activities and expresses her own opinions; users who have questions could source for help in CQA systems and there are CQA systems for question-answers in general domains as well as in specific domains.

UGC data from different platforms share at least three common characteristics: (1) **Huge Amount.** Any user could be a content contributor in Web 2.0 platforms. This mechanism causes the information explosion. For instance, in our study, we collected the most-commented news articles from Yahoo! News and their comments.

The average number of comments is 1059 per news article in our data collection. On Twitter platform, 58 million tweets are posted per day by more than 645 million active Twitter users¹; Stack Overflow, for professional and enthusiast programmers, has over 2 million registered users who posted more than 7 million questions as at April 2014².

(2) **Free Style.** In general, UGC is a type of informal text. On the one hand, UGC includes different languages, abbreviations, emoticons, slang words and many others. For example, due to the length limitation of 140 characters in Twitter, abbreviations and emoticons are widely adopted in tweets to convey more information. On the other hand, UGC data is not restricted by the topics of discussion. When commenting a news article, a user may post something about the background of the reported event in the news article, some related events, and even her personal opinions. (3) **Heterogeneous Data.** UGC data is not just plain text. It includes various types of attributes depending on the specific platform. Common attributes are the user or the creator of the data, the time of creation and the location, textual content and even social connections of the user. For example, a typical tweet has a user name, publish time, tweet content, and some tweets may have hashtags, mentions, links to external resources, and GPS locations.

The key focus of this research is the text attribute in UGC data and their dependency relationship. Depending on the platform within which the data is created, UGC data often demonstrates dependency relationship. For instance, a reader's comment is always associated with a news article; a hashtag is embedded in a tweet; and an answer belongs to a question. We call the news article, tweet, question examples of **master documents** and its associated comments, hashtags, and answers **slave documents**. Note that one master document might have zero, one, or more slave documents and there might be some other types of relationships among slave documents and/or their creators, *e.g.*, one comment replies another comment of the same news article, a user may vote an answer contributed by another user.

¹<http://www.statisticbrain.com/twitter-statistics/>

²http://en.wikipedia.org/wiki/Stack_Overflow

1.2 Topic Discovery and Slave Document Ranking

Though the UGC data with the master-slave relationship benefits users in quickly and widely posting and aggregating information, it triggers two important issues. (1) The overwhelming information flow of UGC causes the users' reading overload. As aforementioned, there are on average 1059 comments per article among the most-commented articles collected from Yahoo! News in our study. It is time-consuming for users to read a large number of comments in an unorganized manner. (2) The informal writing style of UGC, particularly the large use of slang words and abbreviations could hurt users' understanding. For example, many hashtags in Twitter are abbreviations of one or more words (*e.g.*, `#sg` for Singapore and `#jb` for Justin Bieber). Without contextual information, it is difficult to get the meaning of such kind of hashtags. In our study, on the one hand, we utilize the master-slave relationship of UGC and design topic models to discover fine-grained topics in UGC data. Given a master document, we rank slave documents associated with this master document to promote high-quality slave documents, with the aim of easing users' reading overload and enhancing users' understanding.

Discovering fine-grained topics in UGC helps users in quickly comprehending the information and the opinions conveyed by UGC. However, simply treating UGC data as plain documents inevitably causes the loss of the structure information among the data. For a given news article, we observe that a subset of its comments may discuss the topics mentioned in the news article, but usually not all of its comments are confined to the topics covered by this news article alone. For CQA systems, even the questions and answers are about the same or similar topics, the words used in question topics and answer topics might be different. Take the "Java programming" topic as an example, the words of question topic referred to "Java programming" could be `java`, `eclipse`, and `ide`; while the keywords of answer topic referred to "Java programming" could be `cache`, `compile`, and `request`. Consequently, capturing the relationship between different types of text pieces is the key for discovering fine-

grained topics, and in return the discovered topics help users better understand the UGC data from topic level.

A master document is often associated with multiple slave documents. Given a master document, the huge volume and noise of its slave documents could confuse users' reading and understanding. Selecting and ranking a subset of high-quality slave documents for a given master document is a convenient and practical way for users to read less and get more. (1) The subset of the high-quality slave documents cover more topics and perspectives. For example, the top-ranked comments of a given news article are considered as the summarization of all the comments, which should not only mention the event discussed in the news article, but also talk about the background of the event, related events and the opinions about the event. (2) The subset of the high-quality slave documents represents the master document. Hashtag is often a topical annotation of the tweet containing it. When annotating a tweet with hashtags, the recommended and ranked hashtags should best match the theme of the tweet. (3) The subset of the high-quality answers well addresses the questions. In CQA systems, the answers best addressing the asked question should be promoted. In this research, the above three problems are investigated.

Comment Ranking for News Articles. Due to the large readership, a popular news article (*i.e.*, a master document) may easily accumulate thousands of comments (slave documents) within a short time period. It's unnecessary and time-consuming for users to read all the comments of a news article. Currently, comments can be ranked in a number of ways to ease users' reading overload. For example, comments on Yahoo! News website can be ranked by three criteria: recency, popularity (*e.g.*, based on user ratings), and number of replies a comment has. However, such ranking mechanisms do not offer readers a quick overview of all topics discussed among commenters. More importantly, user-rating based ranking is very likely to suffer from the effect of *rich get richer*. As a result, all top-ranked comments of a news article may be about one dominating topic. Comments about other topics discussed by readers are ranked at much lower positions. We take the first step to study the problem of

comment ranking for news articles. We also call the problem **Topic-driven Reader Comment Summarization (TORCS)**, for the reason that we aim to identify the latent topics among comments contributed by users of an article and subsequently select a small number of most salient comments as summary. The summary gives a concise representation of all comments according to the identified topics. In other words, given a master document, we attempt to select and rank its slave documents, which not only cover topics discussed in the given master document, but also refer to the background and opinions. TORCS benefits users by offering an alternative way of quickly ascertaining overall topics of thousands of comments of an article in addition to the existing comment ranking mechanisms. Moreover, TORCS enables a user to selectively read comments on one specific topic chosen by her from all latent topics identified. We argue that TORCS not only benefits Web users, but more importantly offers great value to mobile users. Due to the relatively limited screen size, mobile users do not have the luxury of reading a large number of comments without scrolling. TORCS brings a more organized way of reading comments through mobile devices.

Hashtag Ranking for Tweet Annotation. In our study, we consider hashtag as the slave document of its corresponding tweet (master document). Hashtag has demonstrated its effectiveness in bringing organization to the sparse information in Twitter. Hashtags associated with tweets enhance information diffusion and tweet search as well as facilitate social chatting. Reported in a recent survey by RadiumOne [Rad13], 58% of Twitter users utilize hashtags on a regular basis. Because of its effectiveness, hashtag has been adopted as a key feature in other micro-blogging services like Tumblr and Sina Weibo, and recently has been officially supported in Google+ and Facebook.³ The effectiveness of hashtags in tweets, however, is limited by the freedom of users in deciding (i) whether or not to annotate tweets with hashtags, and (ii) which hashtags to use among a few alternatives for the same semantic meaning. In 2010, only about 11% of tweets were annotated with one or more hashtags [HCC11]. Our study aims at annotating tweets with the most appropriate

³<http://en.wikipedia.org/wiki/Hashtag>

hashtags. In other words, given a tweet as a master document, we attempt to rank candidate hashtags, which could best annotate the tweet.

Answer Ranking for Questions. In CQA systems for general topics (*e.g.*, Yahoo! Answers, Baidu Knows), users may ask questions of any topic, *e.g.*, “*What is Paris famous for?*”. To answer this question, no much professional knowledge is needed. In domain-specific CQA systems (*e.g.*, Stack Overflow, MedHelp), professional knowledge is required to answer questions like “*How does ruby on rails handle requests?*” from Stack Overflow, and “*How do I know when skipped heart beats are dangerous?*” from MedHelp. While a lot of studies have been carried out on CQA for general topics [XJW12, LSG12, DKMS11], there are limited studies on domain-specific CQA systems. Users who have questions which require professional knowledge are more likely to source for help in domain-specific CQA systems, because these systems enable comprehensive interaction between users on fine-grained domain-specific topics. UGC data from domain-specific CQA systems has an obvious master-slave document structure. The popularity of domain-specific CQA makes a question (master document) receiving many answers (slave documents) within a very short time period. It is difficult for a user to distinguish high-quality answers from low-quality answers, for the reason that the answers may be from a domain the asker is unfamiliar with. Our study benefits users in quickly getting high-quality answers. In other words, we promote slave documents of a given master document, which could best solve the question proposed in the master document.

1.3 Approaches and Methodologies

In order to rank slave documents for a given master document, we first employ topic models to discover topics in both master and slave documents. Then we rank and recommend slave documents based on topical similarity measure. We detail the approaches and methodologies of the three research problems defined above.

Comment Ranking for News Articles. In this task, we assume that topics of slave documents not only cover topics of their corresponding master document, but also refer to sentimental topics from slave documents themselves.

Considering a comment is usually a piece of short text⁴, at first glance, techniques on short text clustering and classification (*e.g.*, search snippets clustering [SH06, SFMC12] and classification [CJS11]) can be directly applied. However, these techniques do not consider the *news-comment*, *comment-comment*, and *news-news* relationships. As defined in Merriam-Webster dictionary, a comment is “a note explaining, illustrating, or criticizing the meaning of a writing” or “an observation or remark expressing an opinion or attitude”. For a given news article, we observe that a subset of its comments may discuss the topics mentioned in the news article, but usually not all of its comments. For instance, a news article talking about tax policy in US may have comments on tax policy, comments on health care reform, or gossip of persons mentioned in the article. That is, comments may extend the discussion with topics related to the news article or even more general topics completely irrelevant to the news article. Furthermore, in a news stream, many news articles reported on different days are on similar topics (or even the same event). For example, many news articles are about the event of US presidential election.

To identify the latent topics in comments with consideration of the relationships between comments and news articles, we propose two LDA-style topic models, namely, ***Master-Slave Topic Model*** (MSTM) and ***Extended Master-Slave Topic Model*** (EXTM). The former assumes that all the topics discussed in comments are derived from the corresponding news articles. The latter is based on our observation that comments may discuss the topics derived from news articles as well as the extended topics not directly derived from news articles. In both models, we assume that each comment has exactly one topic due to the short length of most comments. Such simple topic assignment makes the comment clustering efficient. A cluster is formed by simply grouping comments with the same topic. After cluster-

⁴More than 60% of comments in our data collection has fewer than 10 words each.

ing process, we select representative comments using two methods, namely, Maximal Marginal Relevance (MMR) and Rating & Length (RL). The former is widely used in document summarization to select comments based on topical relevance while avoiding redundancy; the latter considers the user rating and the length of comment for representative comment selection. Users get more comprehensive understanding of the comments by giving topical and distinctive comments than representing summary merely using keywords. In our evaluation, we observe that the proposed topic-driven approach avoids redundancy problem faced by the word-based clustering approach to some extent.

Hashtag Ranking for Tweet Annotation. In this task, we assume that the topics of slave documents are the topical summarization of their corresponding master document. Besides hashtags (slave documents) and tweet content (master documents), we also consider two additional factors: user (*i.e.*, author of a tweet) and posting time of the tweet. As a form of high-level topic abstraction, hashtags in a collection of tweets directly reflect personal activities and hot events in Twitter at that time period. We therefore aim to model the latent topical relationship not only between *tweet content* (master document) and *hashtag* (slave document), but also between *user*, *time*, and *hashtag*.

- *Tweet content.* As an annotation, a hashtag is a high-level abstraction of the content of a tweet. Among all factors, tweet content is the most important factor affecting the usage of hashtags. However, there could be two kinds of possible associations between a hashtag and a tweet: (i) a user composes a tweet and then finds one or more appropriate hashtags to describe the tweet. In other words, before user finishing writing this tweet, she has no particular hashtag in mind to use. A hashtag is chosen because it best describes the tweet content. (ii) a user composes a tweet with a specific hashtag in mind. In this case, the tweet content could be considered as a detailed elaboration of the pre-chosen hashtag or comment on the event indicated by the hashtag. In our

study, we propose two models to model the two different generation processes between tweet content and hashtag.

- *User*. In general, a large portion of tweets from a common user are about her personal interests/activities (*e.g.*, **music**, **sports**, **food**, **travel**). The hashtags adopted by a user often reflect such interests and activities. Some of the common hashtags (*e.g.*, **#nowplaying**, **#nba**) adopted by a large number of users sharing similar interests lead to informal social communities through these common hashtags as well as the mention mechanism. We therefore consider user as a factor in affecting hashtag annotation and also evaluate the impact of social factor on affecting hashtag annotation.
- *Time*. Twitter is a real-time social media. Many of the tweets are about recent or ongoing events. Many tweets, and their associated hashtags, published in a time period are about hot events at that time period. The time factor enables our models to better associate time-sensitive hashtags with tweets.

Considering the three factors and the two generation processes, we propose two PLSA-style topic models, namely, ***Content-Pivoted Model*** (CPM) and ***Hashtag-Pivoted Model*** (HPM), to jointly model the relationship between user, time, tweet content and hashtag, at topic level. CPM assumes that a user composes a tweet and then finds the appropriate hashtags to describe the tweet. HPM assumes that a user composes a tweet with pre-selected hashtag(s) in mind. The experimental results demonstrate the effectiveness of our methods.

Answer Ranking for Questions. In this task, we assume that topics of slave documents and topics of their corresponding master document are similar, but the top words of slave topics and master topics are from different vocabularies. More specifically, in CQA systems, we consider three roles of users, *asker*, *answerer*, and *voter*. Note that, a user may perform the three roles (*i.e.*, asker, answerer, or voter) simultaneously across different questions in a CQA system.

- *Asker*. To ask a question of unfamiliar topics, a user composes the question and waits for answers to this question. Here, an asker is assumed to be lack of knowledge on this question.
- *Answerer*. If a user believes that she has the knowledge to answer this question, she contributes an answer. Here, there is an *implicit* question selection activity where an answerer performs a self-assessment whether she has the knowledge to answer this question.
- *Voter*. In CQA systems, users are often allowed to vote for the answers to a question, based on their judgements. The answers received more votes are therefore believed better addressing the corresponding question.

We propose a PLSA-style topic model, named ***Tri-Role Topic Modeling*** (TRTM), which makes use of the three roles of users for modeling users' activities and for mining fine-grained topics in domain-specific CQA systems. As aforementioned, a user composes questions in her unfamiliar topics, and contributes answers if she believes that she has the right knowledge. Users also vote positively for answers that well address the questions. We therefore argue that the topic distributions of the asker role and the answerer role of the same user could be very different (*e.g.*, unfamiliar topics vs familiar topics). Moreover, if an answer receives a large number of positive votes, then the answer is believed to be of similar topic distribution with the question. TRTM makes three assumptions: (1) An asker and all the questions composed by her share similar topic distributions; (2) An answerer and all the answers contributed by her share similar topic distributions; and (3) An answerer's topic distribution is more similar to that of the questions answered by her, if her answers to these questions receive many positive votes. To evaluate the effectiveness of our model, we apply TRTM to the application of ranking answers for questions. TRTM outperforms two state-of-the-art baselines on real data collected from Stack Overflow on this task.

Our proposed three ranking problems are all based on UGC data from different platforms with the master-slave structure. Depending on the different features of

UGC data, our models extend classic topic models decorated with various features. For the problem of comment ranking, because slave documents (comments) are much shorter than their corresponding master document (news article), we assume that the generation of the master document guides the generation of slave documents to some extent. Our main concern is discovering topics from comments which reflect the topics of their news article as well as keeping topics merely discussed among comments. For the problem of hashtag ranking, the slave documents (hashtags) are extremely short, and in most cases, the hashtag is only in form of one or a few words. Compared with MSTM and EXTM, additional factors like user and time are introduced into CPM and HPM to enrich the representation of hashtags for more accurate ranking. For the problem of answer ranking, since users have multiple roles (*e.g.*, asker, answerer and voter), unlike modeling a single role for the problem of hashtag ranking, TRTM models these three roles as well as the relationships among these three roles. Furthermore, the answers could receive votes from voters. It is very challenging for us to incorporate votes into TRTM. We treat votes as a regularization implemented by the exponential KL-divergence function in TRTM, which better model the relations between questions, answers, askers and answerers. We believe that the more powerful topic models with more features are helpful for us to address our ranking problems.

1.4 Contributions and Thesis Organization

In this research, we focus on topic modeling of User Generated Content. By focusing the master-slave relationship of documents in three types of UGC, we made the following three major contributions.

- We define the problem of comment ranking for news articles (also known as the problem of *topic-driven reader comment summarization*). This research problem aims at ranking a subset of comments (or slave documents) of a news article (master document) which cover the topics of the news article and extended topics of discussions by news readers. To solve this problem, we propose

two LDA-style topic models, Master-Slave Topic Model (MSTM) and Extended Master-Slave Topic Model (EXTM), to integrate the generative process of news articles and that of comments in a unified manner. Both models are based on the observation that topics of news articles have significant effect on topics of comments, and a single comment usually is specific to one topic. EXTM further considers topics that are generated from comments themselves which are not strongly related to news articles, simulating the generative process of news-comments document in a more natural manner. The proposed models are evaluated on a dataset collected from Yahoo! News. Experimental results show that our methods outperform baseline methods. This contribution is reported in Chapter 3.

- We define the problem of hashtag ranking for tweet annotation. This problem aims at ranking a subset of hashtags as slave documents that best summarize their corresponding tweets which are considered as master documents. To solve this problem, we propose two PLSA-style topic models, Content-Pivoted Model (CPM) and Hashtag-Pivoted Model (HPM), to simulate the two hashtag generation processes and both consider three important factors (*i.e.*, user, time, tweet content). Based on the assumption that users who often mention each other are more likely to share similar topics, we further introduce social network regularization into the two models to evaluate the impact of social factor. The experimental results show that our methods achieve superior recommendation accuracy for tweet annotation. This contribution is reported in Chapter 4.
- We define the problem of answer ranking for questions in CQA system. This problem aims at ranking a subset of answers as slave documents best address their corresponding questions which are master documents. To solve this problem, we propose a PLSA-style topic model, Tri-Role Topic Modeling (TRTM), which makes use of the three roles of users (asker, answerer, and voter), for modeling users' activities and for mining fine-grained topics in domain-specific

CQA systems. TRTM outperforms two state-of-the-art baselines on real data collected from Stack Overflow on this task. This contribution is reported in Chapter 5.

Finally, Chapter 6 concludes this thesis and suggests a few promising directions for future research.

Chapter 2

Literature Review

Topic models are a set of algorithms to discover topics from a collection of documents. Due to their flexibility and easy usage, topic models have attracted many researchers' attention from both academic and industrial fields for a long time period. There are generally two classic topic models, namely, Probabilistic Latent Semantic Analysis (PLSA) [Hof99] and Latent Dirichlet Allocation (LDA) [BNJ03]. Both models assume that a document consists of a set of topics and a topic consists of a set of words. Topic models have been applied to various research areas, including topic discovery, document ranking, document classification, opinion mining, etc.. In this chapter, we briefly review PLSA and LDA as well as their extensions and applications.

2.1 Probabilistic Latent Semantic Analysis

The Probabilistic Latent Semantic Analysis (PLSA) is a classic topic model proposed in [Hof99], which defines a generative model for the analysis of text data. PLSA is considered a pioneering probabilistic model for text mining and information retrieval. In this section, we first detail PLSA, including background, generative process, and parameter estimation of PLSA. Then we review the extensions of PLSA applied in various fields. Finally, we describe two PLSA-style topic models which are most relevant to our work.

Figure 2.1: Probabilistic Latent Semantic Analysis

Background of PLSA. Before the prevalence of PLSA, there are two major models in text mining and information retrieval: (1) Vector Space Model [SWY75], and (2) Statistical Language Model [PC98]. These two models, both based on word feature space, are widely adopted in practice (*e.g.*, in search engines). Nevertheless, with the further development of text mining, mining hidden semantics (topics) of text documents is required to better understand the document collection. For this purpose, Latent Semantic Analysis (LSA) [DDF⁺90] using a singular value decomposition was proposed, which can achieve significant compression for document collection as well as discover hidden topics of document collection to some extent. However, LSA has two unavoidable drawbacks: (1) The result dimensions are difficult to explain; and (2) LSA cannot handle polysemy.

Generative Process of PLSA. PLSA is a significant progress for the task of discovering hidden topics, which overcomes the two drawbacks of LSA. PLSA assumes each document is a mixture of topics and each topic is a mixture of words. Formally, in PLSA, a *document* is a sequence of N words denoted by $\mathbf{w} = \{w_1, w_2, \dots, w_N\}$ and the document collection consisting of M documents is denoted by $D = \{d_1, d_2, \dots, d_M\}$. The generative process of each document d is illustrated in Algorithm 1 and the graphical representation of PLSA is shown in Figure 2.1, where z is topic indicator. PLSA is a *bag-of-words* model, indicating that the order of words is not considered in the model. The joint probability of PLSA is shown in Equation 2.1.

$$p(d, w) = p(d) \sum_{z \in Z} p(w|z)p(z|d) \quad (2.1)$$

Algorithm 1: Generative Process of PLSA

```
1 Choose a document  $d$  with probability  $P(d)$ ;  
2 for each word  $w_n$  in document  $d$  do  
3   | Choose a topic  $z_n$  with probability  $P(z_n|d)$ ;  
4   | Generate a word  $w_n$  with probability  $P(w_n|z_n)$ ;  
5 end
```

We observe that given hidden variable z , document d and word w are conditionally independent.

Inference and Parameter Estimation for PLSA. The exact inference algorithm for PLSA is intractable. An Expectation-Maximization (EM) algorithm is applied to infer probabilities (*i.e.*, $p(z|d, w)$, $p(w|z)$, $p(d|z)$, and $p(z)$). The log likelihood of PLSA can be represented as $L = \sum_{d \in D} \sum_{w \in W} n(d, w) \log p(d, w)$, where $n(d, w)$ denotes the term frequency of w in d . EM algorithm has two steps: (1) In E-step, the posterior probabilities for latent variable z are calculated. (2) In M-step, the likelihood is maximized and parameters are updated. The two steps are shown in the following equations, where $R = \sum_{d, w} n(d, w)$ in Equation 2.5.

- In **E-step**,

$$p(z|d, w) = \frac{p(z)p(d|z)p(w|z)}{\sum_{z'} p(z')p(d|z')p(w|z')} \quad (2.2)$$

- In **M-step**,

$$p(w|z) = \frac{\sum_d n(d, w)p(z|d, w)}{\sum_{d, w'} n(d, w')p(z|d, w')} \quad (2.3)$$

$$p(d|z) = \frac{\sum_w n(d, w)p(z|d, w)}{\sum_{d', w} n(d', w)p(z|d', w)} \quad (2.4)$$

$$p(z) = 1/R \sum_{d, w} n(d, w)p(z|d, w) \quad (2.5)$$

Iterating the E-step and M-steps could achieve a local maximum of likelihood. The probabilities $p(w|z)$ are considered as discovered hidden topics, which are useful features for many tasks in text mining.

2.1.1 Applications of PLSA

PLSA as a classic topic model has strong theoretical foundation and can be easily extended. With the prevalence of PLSA, many research tasks employ PLSA to achieve their goals. In this section, we detail the applications of PLSA in spatial data mining, social annotations, and recommendation systems as examples.

Spatial Data Mining. The rapid development of mobile devices and GPS techniques makes spatial information become easily obtainable metadata. For example, in Twitter, the GPS locations can be identified by smart phones and attached to tweets. Many interesting topics are relevant to geo-locations. Two research studies [YCH⁺11, MLSZ06] work on finding geographical topics from GPS-associated documents. In [YCH⁺11], a PLSA-style model, named Latent Geographical Topic Analysis (LGTA), was proposed. LGTA combines PLSA and geo-clustering in one framework. This model assumes that topics are generated from regions instead of documents and each document is associated with a location (a location is sampled from a region). Under this assumption, words close in space are more likely to be from the same region and thus words from the same region are more likely to be from the same topic. Their experimental results show that LGTA has strong abilities to discover geographical topics on Flickr dataset. Mei *et al.* [MLSZ06] proposed Spatiotemporal Theme Model to find spatiotemporal patterns from weblogs. Their model assumes that a word could be sampled from either background model or topics. Furthermore, topics can be sampled from either document or location and time. Based on this mechanism, Spatiotemporal Theme Model is able to discover fine-grained spatiotemporal topics.

Another research work employing PLSA-style model for POI (Point of Interest) recommendation was proposed in [YCM⁺13]. The probabilistic model called W^4 (short for Who, Where, When and What) incorporates user, geographical location, time and short text messages into a unified framework to model users' mobility behaviors. W^4 holds four assumptions: (1) Each user has multiple regions (*e.g.*, home

Figure 2.2: Topic Model of W^4 (short for Who, Where, When and What)

region and work region). (2) The topics of a user are affected by both the user's interest and the region that the user stays. (3) When a user chooses a location to visit, her decision is affected by both her interest and her current region. (4) Different regions and different topics have different word usage preferences. The main contribution of this work is simultaneously discovering users' topics and regions' topics under the time regularization. In addition, the W^4 model can be applied to various tasks, such as location prediction for tweet and location prediction for user. The graphical representation of W^4 is illustrated in Figure 2.2. Note that u represents user, s and t represent day and time, and r and l represent region and location in W^4 . Compared with the two aforementioned spatial data mining tasks using PLSA-style models, W^4 is considered more practical by modeling more factors. Carried on Twitter data set, the experimental results demonstrate that W^4 achieves good prediction accuracy for the task of location prediction for tweet. All above presented research studies evidence the effectiveness of PLSA-style models for spatial data mining.

Social Annotation. Social annotation services are more and more popular with the exploration of UGC data. A traditional method to organizing web resources is predefining a taxonomy and manually classifying web resources into their correspond-

ing categories. However, the traditional method has two unavoidable drawbacks: (1) It's not practical to manually annotate a huge and growing volume of UGC data by experts. (2) The predefined taxonomy is not able to catch the vocabulary change of UGC data. Thus, annotating web resources by the tremendous number of users is a good alternative method. Web users could freely annotate web resources with any word they like. Instead of a hierarchical category representation supported by the traditional method, social annotation enables the building of a flat category representation. Though social annotation overcomes the two previous drawbacks, it also introduces new problems (*e.g.*, ambiguity of semantics). The ambiguity of semantics is caused by a lack of shared taxonomy. In social tagging systems, the same tag might indicate different things by different users, and different tags might also refer to the same semantic meaning. Topic model is a natural choice for analyzing semantics and mining topics from web resources. Next, we review a topic model for clearing the ambiguity of semantics in social tagging systems.

Wu *et al.* [WZY06] proposed a PLSA-style model to explore social annotations on Del.icio.us data. They stated that an annotation typically consists of four attributes: the link to the web resource, tags, the user who annotates the link, and the time when the annotation is created. Their model mainly considers three attributes (*i.e.*, user, resource, tag) and ignores the time attribute. The model assumes that there exists K topics in the whole web resources and a triple of $\langle user, resource, tag \rangle$ is sampled from the hidden topics. More specifically, the probability of generating a triple is defined as $p(u, r, t) = \sum_{i=1}^K p(z_i)p(u|z_i)p(r|z_i)p(t|z_i)$, where u, r, t, z represent user, resource, tag, and topic respectively. The experimental results show that their model is able to group semantically similar tags and distinguish tags with multiple semantics. Generally speaking, using topics discovered from topic models is a good way to address the problem of semantics ambiguity from social annotations.

Recommendation Systems. Recommendation is a hot issue in the area of Information Retrieval for a long time period. The traditional recommendation systems include two mainstream methods: Collaborative Filtering (CF) [SLH14] and Matrix

Factorization (MF) [SM07]. CF is a simple but effective method for item recommendation based on item-item or user-user similarity. MF treats user and item as k -dimension features and decomposes $U - I$ matrix into $|U| * k$ matrix and $|I| * k$ matrix, where U and I denote the set of users and items respectively. MF is reported to achieve high recommendation accuracy. However, both CF and MF cannot handle cold-start problem. In other words, when a new item comes, it is difficult to recommend this new item to appropriate users when employing either CF or MF.

Topic modeling algorithms are used to discover a set of hidden topics in a document collection. Even for a new item or user, topic models can estimate the topic distribution of the new item or user. Topic model therefore becomes a natural way to handle the cold-start problem in recommendation systems. A summary of using PLSA-style models for recommendation can be found in [Hof04]. Compared to dimension reduction methods like MF, PLSA provides a probabilistic semantics for model selection and inference. PLSA-style model is also flexible and scalable. To name a few examples, Thomas [Hof03] presented a generalized PLSA which models user interest groups with Gaussian distributions to implement movie recommendation on EachMovie data; Deng *et al.* [DKL08] addressed the expert-finding task by combining the language model and the topic model on DBLP bibliography data; Schein *et al.* [SPUP02] developed an approach combining Collaborative Filtering and PLSA to solve the cold start problem in recommendation systems on MovieLens data set.

More recent work applying PLSA-style models for the recommendation task are question recommendation on Yahoo! Answers [XJW12] and news article recommendation on Digg data [KPS13]. The former proposed a Dual Role Model (DRM) distinguishing the asker role and answerer role in CQA systems. DRM is able to generate topic distributions of both asker and answerer roles. The graphical representation of DRM is shown in Figure 2.3, where u^q , u^a and q represent asker, answerer, and question respectively. By computing the topical similarity between questions and users, DRM recommends questions best matching users' interest. The authors concluded that the answerer role is more effective for question recommendation and the model

Figure 2.3: Dual Role Model

with two roles outperforms the model with a single role measured by recommendation accuracies on Yahoo! Answers data.

Digg is a social news website which enables users to submit news articles and vote news articles shared by other users. In [KPS13], Kim *et al.* proposed DIGTOBI, a personalized recommendation system for Digg articles, using votes provided by users. The generative process of DIGTOBI consists of two steps: (1) *Writing Digg articles*, and (2) *Digging Digg articles*. A user u decides to write (or submit) an article d with the probability computed by the function $EXP - KL(u, d)$. Note that $EXP - KL(u, d)$ function is controlled by the topical similarity between u and d , which indicates that if the value of the topical similarity is larger, u is more likely to write d . Then, for each word w in d , a topic z is sampled from $p(z|d)$ and consequently w is sampled from $p(w|z)$. A user v decides to dig d or not with the probability computed by the function $EXP - KL(v, s, d)$, where s is the accumulated Digg vote received by d . In this mechanism, a popular news article (a news article receiving a larger number of Digg vote) is less affected by users' interest. In other words, users with different preferences might dig the same popular news articles. DIGTOBI can handle both warm-start and cold-start problems and shows the superior ability

of PLSA on recommendation systems compared to a set of strong state-of-the-art baselines. Our TRTM model for the problem of answer ranking considers both the multiple roles in CQA systems from DRM as well as the vote score from DIGTOBI, which helps us more accurately simulate the users' activities in CQA systems.

PLSA is often considered as a strong technique in text mining. Opinion mining with PLSA [GL11, MLW⁺07, LHAY07, YLHA12] is a popular research direction. In [LHAY07, YLHA12], the authors proposed a Sentiment-PLSA model for predicting sales performance. The Sentiment-PLSA assumes that a document is generated by a number of hidden sentiment factors. A summary of the sentiment information can be obtained by training the Sentiment-PLSA model and this sentiment information is a type of useful feature for sales performance prediction. Other research tasks includes event detection [MZ05] and novelty discovery [AH13].

2.1.2 Extended PLSA Related to Our Work

PLSA is a scalable and flexible model. On social platforms (*e.g.*, Facebook, Twitter), users can be friends with each other to form a social network. Social networks play a key role for analyzing relationships between users. Researchers can easily incorporate social networks into PLSA to discover fine-grained topics in UGC data. We employ PLSA-style topic models for hashtag ranking for tweet annotation, and our models also considered social networks. Next, we review studies combining PLSA and social networks that are relevant to our work.

Mei *et al.* [MCZZ08] proposed a framework with both topic models and social networks for community detection and user-topic analysis. Friends are more likely to share similar topics. Based on this assumption, Mei *et al.* considered social network as a regularization and incorporated this regularization into the likelihood of PLSA. Their model is effective in finding topical communities. A recent work holding the same assumption is reported in [GCXZ11]. Guo *et al.* proposed Regularized Topic Model for intent-aware query similarity. The authors argued that query similarity is relevant to users' search intents. For example, if a user is looking for apple fruits, the

query *apple* is similar to the query *apple tree*; while if a user is looking for products by Apple, the query *apple* is similar to the query *apple store*. The authors employed search snippets and clickthrough data to analyze users' search intents. Formally, they suppose that there are K search intent s 's and a query q consists of words w 's from its snippets. The generative process is illustrated in Algorithm 2. We observe that the generative process is the same as PLSA. The novel part of their work is the consideration of co-clicks between two queries and building a query network (similar to social network). That is, if two queries share many commonly clicked URLs, the search intents of these two queries are similar. The regularization formula is defined as follows:

$$R = \sum_{q_i, q_j \in Q} \sum_{s \in S} C_{i,j} (p(s|q_i) - p(s|q_j))^2 \quad (2.6)$$

where $C_{i,j}$ denotes the co-click number between query q_i and q_j . The authors employed both EM and Newton-Raphson methods to learn the model. Measured by average purity score, Regularized Topic Model outperforms PLSA.

Algorithm 2: Generative Process of Regularized Topic Model

- 1 Choose a query q with probability $p(q)$;
 - 2 Choose a search intent s with probability $p(s|q)$;
 - 3 Generate a word w with probability $p(w|s)$.
-

Our CPM and HPM models also introduce social network regularization for modeling topical relationships between users. Our models are significantly different from Regularized Topic Model from two aspects: (1) Regularized Topic Model is used for computing query similarity on query log data set, while our models are designed for hashtag ranking in Twitter. (2) Regularized Topic Model simply combines classic PLSA with social network regularization, while our models consider more factors (*e.g.*, user, time) that may affect hashtag adoption and the models simulate two generative processes of hashtags. More details of CPM and HPM are described in Chapter 4.

Figure 2.4: Latent Dirichlet Allocation

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) as the most popular topic model was first proposed in [BNJ03], which provides an unsupervised and probabilistic way to discover latent topics in text collection. Due to its promising performance, LDA is widely applied in text mining, information retrieval, natural language processing and other areas. In this section, we first detail the standard LDA, including generative process, parameter estimation, and evaluation of LDA. Then we review several topic models which relax the assumptions in LDA. Applications of LDA are also surveyed. Finally, we present two topic models that are most related to our research.

Generative Process of LDA. In LDA, the definitions of document d , word w and topic z are same as in PLSA. The generative process of each document d is illustrated in Algorithm 3 and the graphical representation of LDA is shown in Figure 2.4. Note that α and β are hyper-parameters for Dirichlet distribution; θ is the Multinomial distribution for topics of a document; $p(\theta|\alpha)$ represents a Dirichlet distribution shown in Equation 2.7. This statistical model reflects the intuition that documents exhibit multiple topics. Given hyper-parameters α and β , the joint distribution of θ , $\mathbf{w}$, and $\mathbf{z}$ is given in Equation 2.8. The generative process of LDA defines a joint distribution over observed variables (*i.e.*, words of documents) and hidden variables (*i.e.*, topic structures). We observe that LDA is a three-level model. The hyper-parameters α and β are corpus-level parameters, which are experimentally determined. θ is a document-level variable, while w_n and z_n are word-level variables.

Algorithm 3: Generative Process of LDA

```

1 Choose  $N \sim \text{Poisson}(\xi)$ ;
2 Choose  $\theta \sim \text{Dir}(\alpha)$ ;
3 for each word  $w_n$  in document  $d$  do
4   | Choose a topic  $z_n \sim \text{Multinomial}(\theta)$ ;
5   | Generate a word  $w_n$  from  $p(w_n|z_n, \beta)$ ;
6 end

```

LDA has three main characteristics: (i) To facilitate model inference, LDA employs conjugate prior distributions. Based on Bayes Theorem, $P(\vartheta|\mathcal{X}) \propto P(\vartheta)P(\mathcal{X}|\vartheta)$, where $P(\vartheta)$ denotes prior distribution of parameter ϑ and $P(\vartheta|\mathcal{X})$ denotes posterior distribution of ϑ given observed variable $\mathcal{X}$. If prior distribution and posterior distribution are in the same function form, they are called conjugate pairs (*e.g.*, Dirichlet distribution and Multinomial distribution are conjugate pairs). (ii) LDA assumes that words are generated by topics and these words are exchangeable within the document. In simple words, LDA ignores the order of word sequence in a document as in PLSA. (iii) Different from PLSA, in LDA, θ is treated as a random variable generated from a posterior distribution with its corresponding hyper-parameters, which does not complicate the model when the size of corpus grows.

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1} \quad (2.7)$$

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^N p(z_n|\theta) p(w_n|z_n, \beta) \quad (2.8)$$

Inference and Parameter Estimation for LDA. In general, the exact inference for LDA is intractable. Variational Inference [BNJ03] and Gibbs Sampling [GS04] are two popular approaches to inferencing parameters of LDA-like models. The former employs Jensen's inequality to simplify the log likelihood, which modifies the original graphical models (*e.g.*, remove some edges or nodes). The latter is a member of algorithms from the Markov Chain Monte Carlo (MCMC) framework [Gil99], aiming

to construct a Markov chain. Gibbs Sampling is based on the conditional distributions of variables and it generates a converged posterior distribution after a number of iterations. Generally, Variational Inference is more efficient, while Gibbs Sampling yields more reliable results. In recent years, with the prevalence of stream data and big data, online LDA [HBB10] and distributed LDA [NASW09] are also proposed.

Evaluation for LDA. Because documents for LDA training have no ground-truth labels and topics cannot be predefined, evaluating the performance of LDA is still a difficult issue. Currently, the evaluation measures can be classified into three categories: (i) perplexity, (ii) discovered topics, and (iii) indirect evaluation. Note that for other topic models like PLSA, these three evaluation measures are also applicable.

Perplexity is an objective measure to evaluate the performance of topic models, which assesses models through their likelihood. Formally, perplexity is defined in Equation 2.9, where $P(\mathbf{w}_d)$ denotes the generating probability of words from document d and N_d denotes number of words in document d . A newly proposed topic model is usually compared with standard LDA on the same dataset by perplexity and lower perplexity indicates better performance.

$$perplexity(D_{test}) = \exp \frac{-\sum_d \log P(\mathbf{w}_d)}{\sum_d N_d} \quad (2.9)$$

Recall that topics can be represented by mixture of words and each word w has the probability $P(w|z, \beta)$, we can rank words in each topic based on the probability $P(w|z, \beta)$. The generated topics will be shown to users and users evaluate whether these topics are meaningful or not (*e.g.*, assessing whether words with highest probabilities indeed belong to the same topic). Additionally, each document can be represented by mixture of topics, and two or three topics with highest probability $P(z|\theta_d)$ are usually selected to describe document d . Users are required to evaluate the relation degree between document d and its selected topics.

Indirect evaluation is widely used in application-oriented topic models. After generative process of topic model, document-topic vector and topic-word vector are

obtained. These two vectors can be incorporated in document similarity or topic similarity. For instance, document-topic vector as topic-level feature to enrich short text representation is used to improve performance of short text classification.

Currently, there is still lack of formal theory to evaluate the performance of LDA and other topic models. A recent study [TMN⁺14] winning ICML14 best paper award works on systematic analysis of the performance of topic models. In this study, the authors justified several factors affecting model performance, including the number of documents, the length of individual documents, the number of topics, and the hyper-parameters. It shows that the number of documents probably is the most important factor affecting the performance of topic models.

2.2.1 Relaxing Assumptions of LDA

Though standard LDA is powerful to discover hidden topic structures of a document collection, the assumptions in LDA might hurt the model performance when confronted more specific and complicated requirements (*e.g.*, discovering more sophisticated topic structure of document collection). How to relax these assumptions is a popular research direction of topic models. We review several studies on relaxing assumptions of LDA in the following.

The standard LDA does not consider the order of words in one document. Nevertheless, this assumption is not reasonable in some cases. Intuitively, words close to each other are more likely to refer to the same topic. Consequently, the assumption of exchangeability of words possibly adds more noise to the generated topics. A number of extensions focus on modeling nonexchangeable words. Wallach in [Wal06] presented a Bigram Topic Model assuming that the generative topic of the current word depends on the previous word. A N-gram Topic Model considering topical phrases is proposed in [WMW07]. In the N-gram Topic Model, for each word, a topic is first sampled, then its status (*i.e.*, a unigram or bigram) is sampled, finally the word is sampled from a topic-specific unigram or bigram distribution. Generally, considering the order of words makes models generate more meaningful topics.

The order of documents is also ignored in the standard LDA. This assumption might not be realistic when analyzing data set spanning a long period (*e.g.*, several years or even several centuries) because of topics evolution. Dynamic Topic Model (DTM) [BL06] incorporating the temporal feature into LDA captures topics evolution. DTM describes one topic as a sequence of word distributions based on predefined time interval. An example about the topic of neuroscience is provided in their paper. In 1900s, keywords for neuroscience are *brain*, *eye*, *movement* and *right*. While in 2000s, keywords for neuroscience become *neuron*, *active*, *brain* and *cell*. Note that DTM requires that time should be discretized into intervals. The further extension of DTM called continuous DTM (cDTM) [WBH08] considers time to be continuous. Moreover, cDTM is more efficient to fit the data. Several other topic models considering the temporal feature are proposed in [WSW07, NCD⁺07].

For LDA, the number of topics is experimentally determined. But this assumption might not fit the data. Bayesian nonparametric topic model using Hierarchical Dirichlet Process (HDP) [TJBB06] provides a more natural way to generate topics. HDP assumes that the number of topics is an unknown prior and to be inferred from the data. Additionally, a new document has the opportunity to obtain previously unseen topics. Two popular representations of HDP are the stick-breaking process and Chinese restaurant process. HDP is more powerful in modeling the topic structures of document collection. Several applications of HDP have been proposed in [CHA⁺11, Cow04, ZSZL10].

The standard LDA also assumes that topics are organized in a flat (*i.e.*, not hierarchical) way, which possibly loses information of topic structures of document collection, *e.g.*, topic on computer science and topic on machine learning belong to different topic levels. To overcome this shortcoming of LDA, the nested Chinese restaurant process (nCRP) [BGJ10] aims to generate infinitely deep and infinitely branching topic trees. nCRP defines a hierarchical topic tree with the constraint that general topics (*e.g.*, computer science) appear near the root and specialized topics (*e.g.*, machine learning) appear near the leaves.

2.2.2 Applications of LDA

Due to its flexibility, LDA has been applied to almost all areas of text mining, information retrieval and natural language processing. For many text mining tasks, UGC data contains metadata (*e.g.*, links, geographical locations, authors). In order to account for the additional information of UGC, many extensions have been proposed to fit such semi-structured data. We survey the following areas network analysis, opinion mining, spatial data mining, summarization and recommendation systems using topic models as examples.

Network Analysis. The network structure widely exists in document collection. *e.g.*, citations between research papers, links between Web documents. Especially for UGC data, there is a huge social network behind these social texts. The network structure facilitates topic models to better interpret semantics of document collection and discover hidden topics. On the other hand, topic models with network structure are also able to discover topical community.

Topic models incorporated with the network structure of documents generate more effectively meaningful and distinctive topics. Relational Topic Model (RTM) [CB09] models the link between two documents as a binary variable. RTM assumes that if two documents have a link between them, topic distributions of these two documents are more similar. The generative process of RTM is shown in Algorithm 4. Note that ψ is the link prediction function (*e.g.*, Sigmoid function) and the main difference between LDA and RTM is that RTM samples the binary link indicator between documents. RTM summarizes a network of documents and predicts links between them. Sun *et al.* [SHGY09] proposed a topic model considering both text and structures of documents, namely, iTopicModel. On the top layer of iTopicModel, the authors defined a Markov Random Field (MRF) to model the relationships among documents. On the bottom layer of iTopicModel, it follows LDA to generate text documents.

Given a large-scale linked document collection, another fundamental problem is to discover and analyze the community formed by authors of documents. The Author-

Algorithm 4: Generative Process of Relational Topic Model

```
1 for each document  $d$  do
2   | Choose  $\theta \sim Dir(\alpha)$ ;
3   for each word  $w_n$  in document  $d$  do
4     | Choose a topic  $z_n \sim Multinomial(\theta)$ ;
5     | Generate a word  $w_n$  from  $p(w_n|z_n, \beta)$ ;
6   end
7 end
8 for each document pair  $d, d'$  do
9   | Generate binary link indicator  $y \sim \psi(\cdot|z_d, z_{d'})$ 
10 end
```

Topic Model [RZGSS04] is an early attempt for this task. It is based on the hypothesis that each author is attached by a topic distribution; each document with multiple authors is attached by a topic distribution; each word in the document is attached to an author. Gibbs Sampling is employed to estimate the topic and author distributions. When obtaining topic distributions of authors, authors are grouped based on their topical similarity. Community-User-Topic (CUT) model [ZML⁺06] extends Author-Topic Model by incorporating hidden semantic community formed by users with similar interests and topics. Other related studies on discovering community with topic model are presented in [LNMG09, SCFS12, YJCZ09].

Opinion Mining. The rapid accumulation of resources with abundant personal opinions (*e.g.*, microblogs, reviews of products) inspires many researchers to study the opinion hidden in the data. The fundamental task for opinion mining is extracting opinion and sentiment from UGC, then summarizing social texts based on their opinion, or constructing sentiment word dictionary. Two types of words, sentiment word (*e.g.*, great, wonderful) and topic word (*e.g.*, room, food), are considered. Items to be commented usually have multiple aspects which could be discovered by topic models. One approach is extracting aspects consisting of both sentiment words and topic words. Another approach is separately extracting aspects with only topic words and their corresponding sentiment words.

Titov *et al.* [TM08] presented Multi-grain Topic Model to extract aspects from online reviews by extending LDA with multi-grain topics. Note that their model does not distinguish sentiment words and topic words of each aspect. The authors observed that LDA can only mine global properties of objects (*e.g.*, product type) and fail to discover more specific aspects of objects. On the other hand, by introducing sentence layer in LDA, Multi-grain Topic Model has the powerful ability to generate both global topics and local topics. Similar research work is reported in [LH09]. Joint Sentiment/Topic Model (JST) aims to detect sentiment and topics simultaneously from reviews by adding an additional sentiment layer between document layer and topic layer. JST is a hierarchical topic model with four layers. In both [TM08] and [LH09], the authors try to add a hidden layer to LDA with the purpose of discovering more specific aspects of objects.

Spatial Data Mining. Spatial information is often attached to UGC. For example, location information can be added to Flickr images and tweets. Users favor showing their current positions when taking a photo or updating a status. Intuitively, topics of the same region are more similar. Utilizing spatial information to capture geographical topics attracts many researchers' attention. Sizov [Siz10] presented GeoFolk Model integrating geodata (spatial coordinates) with Flickr image tags for better content management. In GeoFolk Model, the word generated from the Multinomial distribution and the spatial coordinates generated from the normal distribution are simultaneously sampled. Experimental results show that GeoFolk Model outperforms LDA in discovering geographical topics. Spatial information also appeals Twitter users' attention. Hong *et al.* [HAG⁺12] proposed a unified framework to model geo-tagged tweets in Twitter. They constrained that words in a tweet depend on topic and location of tweets and topics appear with different chances in different regions. In their models, all locations are divided into latent regions. When generating a tweet, the region and location are first selected, then a topic is chosen depending on the latent region.

A recent study on POI (Point of Interest) recommendations using topic models has been proposed in [YCZ⁺15]. In this work, user, POI, time, and words from Foursquare are modeled in a probabilistic framework to reveal mobility behavior of a user. In order to avoid setting topic number, a non-parametric topic model based on HDP, namely Enhanced W^4 (Who, Where, When and What), or simply EW^4 is proposed. EW^4 is based on the following assumptions: (1) Individual user has multiple regions (*e.g.*, home region and work region). (2) The chance that a user stays at a given region is affected by time. (*e.g.*, users are more likely to stay at the home region on weekends.) (3) At different places, users perform different activities. The experimental results show that EW^4 outperforms several state-of-the-art baselines for POI recommendation.

Summarization. Summarization is one of the most common tasks in information retrieval, often involving in extracting the most significant sentences or keywords given the document collection. For instance, query-based summarization tries to generate an informative summary from a document collection for a given query. We review two studies, Bayesian Query-Focused Summarization Model [DM06] and query Latent Dirichlet Allocation (qLDA) [TYC09], to give an overall picture of summarization using topic models. The former assumes that a number of relevant documents are known to the query and sentences in a document are relevant to the query. More specifically, sentences are assigned *degree* to the query (*e.g.*, this sentence is 60% about the query). Extracted ranking sentences based on their relevance *degree* form the summary of document collection for a specific query. The latter incorporates the query information into topic model and models the document cluster with the constraint that the topic distribution is close to the query. Based on the learned qLDA, the importance score of each word are calculated. The informative summary consists of sentences with highest scores. Our MSTM and EXTM models also concern the summarization of documents. However, previous studies do not consider the master-slave structure of UGC data, while we first analyze this relationship of Yahoo! News data and incorporate this relationship into our models for better topic generation.

Algorithm 5: Generative Process of Collaborative Topic Modeling

```
1 for each user  $i$  do
2   | Sample user vector  $u_i$  from  $N(0, \lambda_u^{-1} I_K)$ .
3 end
4 for each item  $j$  do
5   | (1)Sample topic distributions  $\theta_j$  from  $\text{Dirichlet}(\alpha)$ .
6   | (2)Sample item latent offset  $\epsilon_j$  from  $N(0, \lambda_v^{-1} I_K)$  and set the item vector
   |  $v_j = \epsilon_j + \theta_j$ .
7   | (3)for each word  $w_{jn}$  do
8   |   | (i)Sample topic  $z_{jn}$  from  $\text{Multi}(\theta)$ . (ii)Sample word  $w_{jn}$  from
   |   |  $\text{Multi}(\beta_{z_{jn}})$ .
9   | end
10 end
11 for each user-item pair  $(i; j)$  do
12   | Sample the rating  $r_{i,j}$  from  $N(u_i^T v_j, c_{ij}^{-1})$ .
13 end
```

Recommendation Systems. Many studies [Hof04,KFN09,WB11,AC10] have been proposed to utilize extensions of LDA for recommendation, especially for the textual item recommendation. The most remarkable one is Wang and Blei’s work [WB11], which wins the KDD11 best student paper award. In this study, the authors combined LDA and MF in a unified framework and proposed Collaborative Topic Modeling (CTM) for recommending scientific papers. They stated that recommending newest scientific papers is important for researchers to obtain the research trend and this task could not be achieved by simply using LDA or MF. In CTM, the user and item are treated as k -dimensional vector u_i and v_j as in MF. The novel part of CTM is that v_j is the combination of θ_j from LDA and ϵ_j from MF. The generative process of CTM is represented in Algorithm 5, where r denotes the rating. Their experimental results show that CTM makes good predictions on unrated papers and CTM offers interpretable user profiles. Introducing topic models into recommendation systems is a big progress for textual item recommendation, for the reason that textual item has additional semantic information and topic model is a powerful tool to mine semantic information.

LDA is an emerging field in machine learning. Other tasks using LDA include information extraction [WL04, ZZSW09], spam filtering [BSSB09, BSB08] and event detection [DJZL12, LZMH10].

2.2.3 Extended LDA Related to Our Work

We employ LDA-style topic models for ranking comments for news articles and for ranking answers for questions in CQA system. For the former task, we extended LDA to fit structures of our data (master-slave documents). We present two existing topic models most relevant to this problem. For the later task, we employ Topic Expertise Model (TEM) [YQG⁺13] as a state-of-the-art baseline, which is reviewed shortly.

Most germane to the comment ranking study are the correspondence LDA (corrLDA) model [BJ03] and the topic-perspective model [LHC⁺10]. These two models are applied to the problem of social annotation, which aims to recommend tags to various items (*e.g.*, documents, images). In the corrLDA model (see Figure 2.5, where r denotes a region of an image), it assumes the word topic of the document (*i.e.*, the item) samples the tag topic, which indicates that the strong relations between word topics and its corresponding tag topics can be captured by corrLDA. This rigid assumption, however, is relaxed in topic-perspective model. The model assumes that user perspective is also able to sample tags. For example, some tags (*e.g.*, **happy**, **delicious**) reflect perspective of users themselves, but not their annotated items. In order to control where a tag is generated from (*i.e.*, document topic or user perspective), a switch is introduced to the model. Each user has a user perspective distribution to describe her preference and interest.

However, because of the significant differences between tags and comments, these two models cannot be applied directly to comments. First, the primary motivations for tagging are organizational and social [MNBD06]. Hence, in social tagging systems, a user often selects or defines a set of keywords to annotate her items (*e.g.*, documents, images). Naturally, each tag leads to multiple topics derived from its set of annotated documents. While a comment is used for a user to express her opinions

Figure 2.5: correspondence Latent Dirichlet Allocation (corrLDA)

on a particular article without organizational motivation. Because of the shortness of most comments, they often have one topic. In addition, due to that a large number of comments are from anonymous users, compared to social tagging systems, the social motivation for commenting is much weak. Many users prefer to voice out through commenting without releasing their identity.

Our proposed MSTM model is similar to corrLDA, but we constrain that words from a single comment share the same topic derived from the news article (*i.e.*, the master document). The same constraint is imposed to the proposed EXTM model. However, different from MSTM, EXTM employs a switch to allow a comment to have topic derived from not only the news article but also comments themselves. The switch in our model is at document-level not at tag-level as in topic-perspective model. Another major difference between EXTM and topic-perspective model is that in EXTM we do not model user information due to the aforementioned reason.

Most germane to our model for ranking answers for questions is the Topic Expertise Model (TEM) proposed in [YQG⁺13]. TEM jointly models topics and expertise of users in a unified framework for recommending experts or answers for given questions in CQA systems. Gaussian mixture hybrid is used to model votes. A vote is a common attribute in CQA system, but rarely employed in previous work. The generative process of TEM is shown in Algorithm 6, where e represents expertise. The novel

Algorithm 6: Generative Process of Topic Expertise Model

```
1 for the  $u$ -th user do
2   | Draw user topic distribution  $\theta_u$  from  $\text{Dirichlet}(\alpha)$ .
3 end
4 for the  $e$ -th expertise do
5   | Draw expertise-specific vote distribution  $N(\mu_e, \sigma_e)$  from  $NG(\alpha_0, \beta_0, \mu_0, \kappa_0)$ .
6 end
7 for the  $k$ -th topic do
8   | (1) Draw word distribution  $\varphi_k$  from  $\text{Dirichlet}(\gamma)$ .
9   | (2) for the  $u$ -th user do
10    | Draw user topical expertise distribution  $\phi_{k,u}$  from  $\text{Dirichlet}(\beta)$ .
11    end
12 end
13 for the  $u$ -th user do
14   | for the  $n$ -th post do
15     | Draw topic  $z$  from  $\text{Multi}(\theta_u)$ .
16     | Draw expertise  $e$  from  $\text{Multi}(\phi_{k,u})$ .
17     | Draw vote  $v$  from  $N(\mu_e, \sigma_e)$ .
18     | for the  $l$ -th word do
19       | Draw word  $w$  from  $\text{Multi}(\varphi_z)$ .
20     | end
21   | end
22 end
```

part of their work is combining topics and expertise of users. More specifically, TEM assumes that a user has a topic distribution, and her topic has expertise distribution representing how the user is expert at the topic. Based on a large volume of question and answer posts as well as votes received by answers, their method can find expert users and rank answers given a new question. The experiments are conducted on Stack Overflow data, and the results evidence the effectiveness of the modeling of votes.

TRTM is significantly different from TEM because TRTM considers and models the three roles of users. Our experimental results confirm that modeling three roles of users benefits mining fine-grained topics of users. TEM was evaluated on three applications: expert finding, similar question searching, and ranking answers for questions. Nevertheless, the problem definition of ranking answers is different from

ours. In [YQG⁺13], only answers from answerers that appear in training data can be ranked. In our problem definition, answers from new answerers can also be ranked, which significantly enhances the applicability of our proposed solution in real-world settings.

In summary, in order to address the three ranking problems, we extended classic topic models to fit the structures and features of our UGC data. MSTM and EXTM for the comment ranking task are based on the LDA model. CPM and HPM for the hashtag ranking task and TRTM for the answer ranking task are based on the PLSA model. As presented in previous related work, topic model is a powerful tool for discovering fine-grained topics in UGC data collection and these topics are very useful features to enhance the performance of various information retrieval and data mining tasks. Previous studies mainly concern modeling the features of UGC data, while our research focuses on modeling both the features and their relations simultaneously. We believe that the relation between features plays an important role in simulating the generation of data set and is helpful for the ranking problems.

Chapter 3

Comment Ranking for News Articles

In this chapter, we address the problem of comment ranking for news articles. We also call the problem Topic-Driven Reader Comment Summarization (TORCS) because our target is to summarize a subset of highly-ranked comments of the given news article to ease users' reading overload. To solve the problem, we propose two LDA-style topic models, Master-Slave Topic Model (MSTM) and Extended Master Slave Topic Model (EXTM). The former model regularizes that the topics of slave documents must be strictly sampled from its corresponding master document. The latter model relaxes the regularization in MSTM and assumes that the topics of slave documents can be sampled from not only its master document but also the slave documents themselves. Both models are used to group comments into topic clusters. Two ranking mechanisms, Maximal Marginal Relevance (MMR) and Rating & Length (RL), respectively, are employed to select a few most representative comments from each comment cluster, to form the comment summary. To evaluate the two models, we conducted experiments on 1005 Yahoo! News articles, which are associated with more than one million comments. Our experimental results show that EXTM significantly outperforms MSTM by perplexity. Through a user study, we also confirm that the

comment summary generated by EXTM achieves better intra-cluster topic cohesion and inter-cluster topic diversity.

The rest of this chapter is organized as follows. Section 3.1 briefly review the related work. Section 3.2 describes TORCS problem, our models (MSTM and EXTM) and methods for comment selection. Section 3.3 extensively evaluates the proposed models measured by perplexity and the discovered topics. Section 3.4 reports the user study. Section 3.5 summarizes this chapter.

3.1 Related Work

Being the most prevalent type of UGC, comments have gained significant interest from researchers for applications on various platforms. Depending on the type of web resources being commented, we classify existing studies into two groups: comments on non-textual resources (*e.g.*, photo and video), and comments on textual-resources (*e.g.*, news article and blog post).

For non-textual web objects like images and video clips, content-based search that solely relies on low-level features (*e.g.*, color, texture, and shape) has been a challenging research issue for a long time [JS10]. Only very recently, progress has been made with the development in the deep learning field. Nevertheless, comments contributed by users are valuable resource to describe these non-textual resources for searching and other applications, because the comments directly record the high-level concepts perceived by human being. For the task of aesthetic photo search, comment-based approach outperforms visual content-based approach [SPYO12]. By incorporating comments, it is reported that video search accuracy is improved by 15% [YYLF09]. A novel topic model was also proposed in [YGC⁺13] to connect tags and comments for images on Flickr. Their experimental results show the effectiveness of using comments for social tagging systems. A detailed analysis of YouTube video comments is reported in [SCNSP10]. The focus of the analysis is on the relationship between the content of the comments and user ratings. The Non-negative Matrix

Factorization method proposed by He *et al.* [HKXC14] clusters songs from Last.fm using comments. The authors showed the promising performance of using comments.

The work reported in [KCH11] on clustering YouTube comments by using topic models is most similar to our research. In [KCH11], all comments of each YouTube video are used to learn a topic model. The comments are then clustered based on the topics assigned by the topic model. Within each comment cluster, a PageRank-style algorithm is used to select a subset of comments as summary. In order to apply the PageRank algorithm, a graph is constructed where each comment in the cluster is a node and an edge is created if two comments share at least m common words (m is a predefined parameter). In our problem setting, although we share similar objective as in [KCH11] to select a subset of comments for each news article, we consider more than the news article and its associated comments. The reason is that many news articles from a news stream may be related to the same ongoing event. In our proposed model, we therefore consider both the topic distribution in comments and the topic distribution in news articles. The two models proposed differ in how the topics in news articles are used in modeling the topics in comments. In other words, our models consider three kinds of relationships between news and comments: news-news, news-comment, and comment-comment relationships. Once assigned to topic cluster, we use two methods to select the most representative comments from each cluster as summary. Both our methods are not based on PageRank.

For studies on comments associated with textual web resources, our research is mostly related to the study of comments on blog posts, because blog posts share lots of common attributes with news articles. A large-scale analysis of blog comments was done by Mishne and Glance in [MG06]. This analysis showed comments are useful in improving the recall measure in blog search. Comments of blog posts also lead to better ranking of blog search results. The work reported in [HSL08] aims to summarize blog posts by identifying the important sentences from them. The sentences are evaluated by the importance of the words contained in them and the word importance is computed based on comments. Three types of relations among

comments are considered in word importance computation: mention relationship, quotation relationship, and topic relationship. Here the topic relationship is evaluated by using cosine similarity of word feature vectors of comments and not topic model. In our proposed topic models, we model the relationships between comments and news articles in a probabilistic framework. Another study on comments for textual web resource is the prediction of the number of comments that a Facebook thread might receive in future [BKLDNM13].

Other research on comments of news articles include spam detection [KSK12], extraction of discussion structure [SMdR07], volume prediction [TWdR09], rating prediction [KHC09], news recommendation based on comments [SKKL12], emotion tagging for comments [ZZS⁺14] and ranking of comments [DSS12]. In [SKKL12], the authors proposed a collaborative filtering approach for personalized news recommendation. In their solution, ratings and features derived from social networks are considered in the recommendation framework. Dalal *et al.* proposed a method to rank comments with multiple objectives [DSS12]; example objectives are ratings, commenter reputation, and recency. There are also studies on the comments on product review. Such studies mainly focuses on sentiment analysis and identification of product features and aspects from comments [WLZ10, DLP03, JO11].

Lastly, our research is also related to short text clustering (see [CORW09] for a detailed survey). In short text clustering, the key issue is to measure the similarity between two pieces of short texts for the reason of word sparsity. To partially address the word sparsity issue, one common approach is to enrich short text representation by using external knowledge bases like WordNet and Wikipedia [CJS11, HSZC09, PNH08]. Topic model based approaches have also been proposed. In [CJS11], latent topics at multiple granularity levels are used to enrich short text representation through topic modeling. The authors in [SFMC12] presented a graph based approach for short text clustering. In this approach, each short text is represented by a graph of topics. The clustering of short text is achieved by graph-cut. By utilizing topically related long texts as the auxiliary data, a transfer learning approach proposed in [JLZ⁺11] is used

Figure 3.1: Topic-driven reader comment summarization

to cluster tweets. In the studies on short text, one major assumption is that the short texts are independent with no relationship among them. In our problem setting, we consider not only short text (*i.e.*, comments) but also long text (*i.e.*, news articles). More importantly, news articles and comments demonstrate master-slave relationship since comments are added by readers of the news articles. For the same reason, there is also a dependency relationship between the topic distributions of news articles and that of their comments.

3.2 Comment Ranking and Selection

We consider the problem of comment ranking and selection a ***Topic-driven Reader Comment Summarization*** problem (TORCS). Given news articles and their associated comments, for each news article, our task is to identify a subset of comments in the form of topic clusters satisfying two requirements: (i) comments in each cluster are on one major topic, and (ii) each comment cluster is represented by a few salient comments. As the results, users only need to browse the salient comments organized in topical clusters. This task is different from most existing problem defini-

tions in document summarization. Document summarization is usually achieved by extracting a set of sentences from the given document(s). In our problem definition, we aim to extract a subset of comments from all comments associated with a news article. Based on the above description, there are two major issues to address: (i) identification of the topics discussed in comments, and (ii) identification of the most representative comments within each topic.

Because of the master-slave relationship between news articles and comments, we also refer to a news article as a *master document* and its associated comments *slave documents*. Note that, in this chapter we do not consider the reply relationship. It is a part of our future work to incorporate reply relationship in our models. Therefore, we consider a master document and its associated slave documents a unified document with star structure, shown in Figure 3.1.

The most straightforward approach to identify the topics discussed among the comments of a given news article is to cluster the comments based on their content similarity (*e.g.*, cosine similarity over word features of the comments). However, this approach simply ignores the context information of the comments and the news articles. Many news articles received from a news stream may be related to the same event¹. Accordingly, the comments associated with such news articles are related at topic level. In the following, we discuss the three kinds of relationships between news articles and comments: *news-news*, *comment-comment*, and *news-comment* relationship. The three kinds of relationships are also illustrated in the top left corner in Figure 3.1.

News-news: Usually a series of news articles are reported for a major event. News articles may also refer to recently published articles for similar types of events (*e.g.*, traffic accidents, natural disaster). We consider two news articles are related to each other at topic level if both news articles are about the same event (*e.g.*, USA presidential election).

¹Grouping news articles about the same event is a major research issue in topic detection and tracking (TDT) [All02].

Comment-comment: Comments associated with the same news article are related through *intra-comment-comment* relationship if the comments are about similar topics. Also, readers may extend the discussion in comments to refer to the event(s) reported in earlier published news articles and the article she is commenting on. In this case, such comments are considered to be related through *inter-comment-comment* relationship. For example, a reader who supports a particular candidate/party in an election may comment on multiple news articles to express her support. Such comments are strongly related to each other but may not be semantically related to the main topic(s) of any particular news article.

News-comment: Similar to comment-comment relationship, there are two types of news-comment relationships: *intra-news-comment* and *inter-news-comment* relationship. The former relates a comment and its associated news article, and the latter relates a comment of one news article to another news article. A user may comment on a news article by referring to information reported in another news article.

Because of the three types of topic-level relationships, to accurately identify the topics discussed among the comments, all these relationships need to be considered. That is, considering only the comments is inadequate. In the following, we present two topic models for the identification of the topics from both news and comments. The two topic models differ in the assumption on the sampling of topics in comments. Given the comments of a news article which are to be summarized, by using the proposed topic model, each comment is assigned a topic cluster. The representative comments are then selected from each topic cluster to form the comment summary. The lower part of Figure 3.1 illustrates the summarization process.

Next, we present the two topic models, namely, Master-Slave Topic Model (MSTM) and Extended Master-Slave Topic Model (EXTM). Figure 3.2 gives the graphical representations of the two models and Table 3.1 lists the notations. With respect to the relationships between news articles and comments, MSTM models the news-news, intra-news-comment, intra-comment-comment relationships, and EXTM models all the relationships discussed earlier.

Table 3.1: Notations for MSTM and EXTM

D, C	Number of master documents, of slave documents
N_d, N_c	Number of words in master document d , in slave document c
V, U	Vocabulary size of master documents, of slave documents
w, e	Word in master document, in slave document
E_c^d	Set of words in slave document c of master document d
N_{dc}^e	Number of times word e occurs in slave document c of master document d
K, F	Number of master topics, number of extended topics
z^m, z^s, x	Master topic, slave topic, and extended topic
θ^m, θ^s	Multinomial distribution over master topics, over extended topics
α^m, α^s	Dirichlet prior vector for θ^m , and θ^s
ϕ_k, ξ_k, ζ_f	Multinomial distribution over words of master document for master topics, over words of slave document for slave topics, over words of slave document for extended topics
β, γ, η	Dirichlet prior vector for ϕ_k, ξ_k , and ζ_f
H	Switch to decide whether the topic is a slave topic or an extended topic
λ, a, b	Probability of topic of slave document being slave topic, and beta prior for λ
$Q_{kd, \neg i}^{KD}$	Number of times master topic k is assigned to master document d , excluding the current word w_i
$Q_{vk, \neg i}^{VK}$	Number of times word v of master document is generated from master topic k , excluding the current word w_i
$Q_{uk, \neg c}^{UK}$	Number of times word u of slave document is generated from slave topic k , excluding all words in slave document c
$Q_{uf, \neg c}^{UF}$	Number of times word u of slave document is generated from extended topic f , excluding all words in slave document c
$Q_{H=1, \neg c}^{CD}$	Number of times H is 1 in document d , excluding the current slave document c
$Q_{H=0, \neg c}^{CD}$	Number of times H is 0 in document d , excluding the current slave document c
$Q_{fd, \neg c}^{FD}$	Number of times extended topic f is assigned to slave documents of master document d , excluding the current slave document c

3.2.1 Master-Slave Topic Model (MSTM)

The proposed Master-Slave Topic Model assumes that all topics of the slave documents are directly derived from topic distribution of its associated master document. In other words, the generative process of the slave documents solely depends on the generative process of master document. However, due to the different vocabularies used in news articles and comments, different words for news articles and comments could be generated for the same topic. We therefore name the same topic *master topic* and *slave topic* when we refer to the topic in the generative process of

(a) MSTM

(b) EXTM

Figure 3.2: MSTM and EXTM. The dotted line illustrates slave topics are derived from news articles; the bold line illustrates the extended topics are derived from comments.

master and slave documents respectively. This generative process is similar to corLDA [BYT⁺09] with the major difference that MSTM assumes that each comment has exactly one slave topic, because most comments are very short. Among one million comments collected from Yahoo! News, more than 60% of comments are shorter than 10 words and more than 80% of comments each has fewer than 16 words (see Figure 3.5). We also observe from the collected data that most comments are specific to one topic.

Figure 3.2(a) shows the graphical representation of MSTM and the generative process is summarized in Algorithm 7. In MSTM, each master document may have multiple master topics, and the generative process of the master document is the same as that in LDA (Lines 4-7 in Algorithm 7). MSTM generates one topic for each slave document based on the uniform word topic distribution of its associated master document, and then generates all words of slave document given the unique slave topic (Lines 8-11).

The exact inference is intractable. Gibbs Sampling (an approximate approach) is utilized to infer the latent variables. Similar to the Collapsed Gibbs Sampling algorithm [GS04], a Markov chain on latent topics is constructed. At each transaction, a topic of word from master document or a topic of slave document is generated with respect to its conditional probability.

The topic of word from master document is drawn from the following conditional probability.

$$P(z_i^m = k | w_i = v, \mathbf{z}_{\neg i}^m, \alpha^m, \beta) \propto \frac{Q_{kd, \neg i}^{KD} + \alpha^m}{\sum_{k'} Q_{k'd, \neg i}^{KD} + K\alpha^m} \times \frac{Q_{vk, \neg i}^{VK} + \beta}{\sum_{v'} Q_{v'k, \neg i}^{VK} + V\beta}$$

The topic of slave document is drawn from conditional probability:

$$P(z_c^s = k | \mathbf{z}_{\neg c}^m, \mathbf{e}_{\neg c}, \gamma) \propto \frac{Q_{kd}^{KD}}{\sum_{k'} Q_{k'd}^{KD}} \times \frac{\prod_{e_j \in E_c^d, e_j = u_j} \prod_{g=0}^{N_{dc}^{e_j} - 1} (Q_{u_j k, \neg c}^{UK} + \gamma + g)}{\prod_{y=0}^{N_{dc} - 1} (\sum_{u'} Q_{u' k, \neg c}^{UK} + U\gamma + y)}$$

Algorithm 7: Generative process of MSTM

```

1 For each of the  $D$  master documents  $d$ , draw  $\theta_d^m \sim \text{Dirichlet}(\alpha^m)$ ;
2 For each of the  $K$  master topics  $k$ , draw  $\phi_k \sim \text{Dirichlet}(\beta)$ ;
3 For each of the  $K$  slave topics  $k$ , draw  $\xi_k \sim \text{Dirichlet}(\gamma)$ ;
4 for each word  $w_i$  in master document  $d$  do
5   | Choose a master topic  $z_i^m \sim \text{Multinomial}(\theta_d^m)$ ;
6   | Generate a word  $w_i \sim \text{Multinomial}(\phi_{z_i^m})$ ;
7 end
8 for each slave document  $c$  of master document  $d$  do
9   | Choose a slave topic  $z_c^s \sim \text{Uniform}(z_{w_1}^m, \dots, z_{w_{N_d}}^m)$ ;
10  | Generate all words in  $c \sim \text{Multinomial}(\xi_{z_c^s})$ ;
11 end
    
```

3.2.2 Extended Master-Slave Topic Model (EXTM)

Extended Master-Slave Topic Model simulates the generative process of both master document and slave document in a more natural way. MSTM follows a rigid constraint that all topics of slave documents have to be derived from the topic distribution of their master documents. EXTM relaxes this rigid constraint and allows some slave documents to have topics irrelevant to their master documents. Such type of topic is known as *extended topic*. As discussed earlier, when commenting on an articles, a reader may provide some additional information (*e.g.*, background information about a news article) which is not covered in the news article. The topics of such additional information are not covered in the original news. In EXTM the notion of extended topic is introduced to take care of such topics.

The graphical representation of EXTM is shown in Figure 3.2(b) and the generative process is summarized in Algorithm 8. In EXTM, the generative process of master document is the same as that in MSTM (Lines 5-8 in Algorithm 8). Although each slave document has one unique topic, the topic can be derived from either the topic distribution of its associated master document, or the extended topic distribution of all comments. A switch H is used to decide whether the topic is a slave topic or an extended topic (Line 11 and 15). If H equals 1, words of slave document are drawn

from the slave topic sampled from uniform word topic distribution of its associated master document (Lines 12 and 13); if H equals 0, words of slave document are drawn from the extended topic sampled from extended topic distribution θ^s (Lines 16 and 17).

Algorithm 8: Generative process of EXTM

```

1 For each of the  $D$  master documents  $d$ , draw  $\theta_d^m \sim \text{Dirichlet}(\alpha^m)$  and draw
   $\theta_d^s \sim \text{Dirichlet}(\alpha^s)$ ;
2 For each of the  $K$  master topics  $k$ , draw  $\phi_k \sim \text{Dirichlet}(\beta)$ ;
3 For each of the  $K$  slave topics  $k$ , draw  $\xi_k \sim \text{Dirichlet}(\gamma)$ ;
4 For each of the  $F$  extended topics  $f$ , draw  $\zeta_f \sim \text{Dirichlet}(\eta)$ ;
5 for each word  $w_i$  in master document  $d$  do
6   | Choose a master topic  $z_i^m \sim \text{Multinomial}(\theta_d^m)$ ;
7   | Generate a word  $w_i \sim \text{Multinomial}(\phi_{z_i^m})$ ;
8 end
9 for each master document  $d$ , draw  $\lambda_d \sim \text{Beta}(a, b)$  do
10  | For each slave document  $c$ , draw a switch  $H \sim \text{Binomial}(\lambda_d)$ ;
11  | if  $H = 1$  then
12  |   | Choose a slave topic  $z_c^s \sim \text{Uniform}(z_{w_1}^m, \dots, z_{w_{N_d}}^m)$ ;
13  |   | Generate all words in  $c \sim \text{Multinomial}(\xi_{z_c^s})$ ;
14  | end
15  | else /*  $H = 0$  */
16  |   | Choose an extended topic  $x_c \sim \text{Multinomial}(\theta_d^s)$ ;
17  |   | Generate all words in  $c \sim \text{Multinomial}(\zeta_{x_c})$ ;
18  | end
19 end

```

Gibbs Sampling is used to infer the latent variables in EXTM. Because the sampling topic of words from master document is the same as MSTM, only the sampling topic of slave document is detailed here. The topic of slave document is drawn from conditional probability $P(H = 1, z_c^s = k | \mathbf{z}_{-c}^m, \mathbf{e}_{-c}, \gamma, a, b)$ when $H = 1$, and drawn from probability $P(H = 0, x_c = f | \mathbf{x}_{-c}, \mathbf{e}_{-c}, \alpha^s, \eta, a, b)$ when $H = 0$.

$$\begin{aligned}
 P(H = 1, z_c^s = k | \mathbf{z}_{-c}^m, \mathbf{e}_{-c}, \gamma, a, b) &\propto \frac{Q_{H=1, \neg c}^{CD} + a}{Q_{H=1, \neg c}^{CD} + Q_{H=0, \neg c}^{CD} + a + b} \times \frac{Q_{kd}^{KD}}{\sum_{k'} Q_{k'd}^{KD}} \\
 &\times \frac{\prod_{e_j \in E_c^d, e_j = u_j} \prod_{g=0}^{N_{dc}^{ej}-1} (Q_{u_j k, \neg c}^{UK} + \gamma + g)}{\prod_{y=0}^{N_{dc}-1} (\sum_{u'} Q_{u' k, \neg c}^{UK} + U\gamma + y)} \\
 P(H = 0, x_c = f | \mathbf{x}_{-c}, \mathbf{e}_{-c}, \alpha^s, \eta, a, b) &\propto \frac{Q_{H=0, \neg c}^{CD} + b}{Q_{H=1, \neg c}^{CD} + Q_{H=0, \neg c}^{CD} + a + b} \times \frac{Q_{fd, \neg c}^{FD} + \alpha^s}{\sum_{f'} Q_{f'd, \neg c}^{FD} + F\alpha^s} \\
 &\times \frac{\prod_{e_j \in E_c^d, e_j = u_j} \prod_{g=0}^{N_{dc}^{ej}-1} (Q_{u_j f, \neg c}^{UF} + \eta + g)}{\prod_{y=0}^{N_{dc}-1} (\sum_{u'} Q_{u' f, \neg c}^{UF} + U\eta + y)}
 \end{aligned}$$

3.2.3 Complexity Analysis

We now analyze the time and space complexity of our proposed MSTM and EXTM. The time complexity for MSTM is $O(T(K|D||N_d^{avg}| + K|D||C_d^{avg}|))$, where T , K , $|D|$, $|N_d^{avg}|$, $|C_d^{avg}|$ are the number of iterations, the number of master topics, the number of news articles, the average number of words per news article and the average number of comments per news article, respectively. Observe that the convergence of our models largely depends on the choices of the number of iterations. The space complexity for MSTM is $O(K|D| + K|V| + K|U|)$, where $|V|$, $|U|$ are the vocabulary size of news articles and comments, respectively. For EXTM, we introduce the extended topics for comments, and the time complexity is $O(T(K|D||N_d^{avg}| + K|D||C_d^{avg}| + F|D||C_d^{avg}|))$ and the space complexity is $O(K|D| + K|V| + K|U| + F|U| + |C_d^{avg}||D|)$, where F is the number of extended topics. Compared with MSTM, an additional time complexity $O(TF|D||C_d^{avg}|)$ is added into EXTM for computing extended topics of comments. Furthermore, $O(|C_d^{avg}||D|)$ is used to store the switch information in EXTM.

3.2.4 Salient Comment Selection

With the assignment of topic labels to comments and each comment is assumed to have one topic only, the clustering of comments is straightforward. The number of comments in each cluster also help to determine the importance (or user focus) of

the topic in user discussion. The next challenge is to select the most representative comments from each cluster to form the comment summary. This task becomes similar to extractive document summarization where the task is to select the sentences from the document to be summarized. Further, comments are short and are similar to sentences. Therefore, the techniques proposed for sentence ranking or selection in document summarization can be easily adopted to address this problem. Two comment selection techniques are evaluated in our study, namely, Maximal Marginal Relevance (MMR) and Rating & Length (RL).

Maximal Marginal Relevance. MMR [CG98] is a widely used technique for selecting sentences in document summarization. MMR aims to select topically relevant sentences and at the same time to reduce redundancy during the selection process. Given all comments in a cluster, its centroid feature vector is firstly computed using the bag-of-word feature representation. The centroid is denoted by c_m . The topical relevance of a comment is then computed based on its similarity to the centroid with the consideration of redundancy against the comments that have already been selected. More specifically, let C_s be the set of comments selected so far. The score of a unselected candidate comment c is computed using the equation below, where $\delta = 0.8$ in our experiments. Note that, the similarity between c and c_m , denoted by $\text{sim}(c, c_m)$, is the cosine similarity, which is widely used for document similarity evaluation.

$$\text{Score}(c) = \delta \times \text{sim}(c, c_m) - (1 - \delta) \times \max_{c' \in C_s} (\text{sim}(c, c'))$$

Because comments are short, $TF \times IDF$ based word weighting scheme is not used in our implementation. Instead, words in comment feature vectors are weighted based on the topic distribution generated from ξ_k or ζ_f . This is reasonable because in topic modeling, each topic is a probability distribution over words, and the higher the probability the more representative of a word on this topic.

Rating & Length. This selection scheme does not consider the content of the comments. Instead, this scheme considers the ratings given by users to comments and also the length of the comments. Rating has always been considered an important feature in comments [HKC09,DSS12]. Yahoo! News website, offers two buttons “like” and “dislike” for readers to evaluate others’ comments. The rating score of a comment is the sum of the likes and dislikes it received. The more likes or dislikes a comment receives, the more attention it gains from the readers. Besides rating, the length of comments in number of words is also used as a feature for comment selection. In [HKC09], the authors reported that length of comments can be used to improve prediction accuracy of comment ranking. To consider both rating and length, the raw values are normalized into $[0, 1]$ by using min-max normalization. The score of a comment in this scheme $Score(c)$ is a linear combination of the two normalized values, given in the equation below, where σ is a parameter for balancing the contribution of the two scores. $\sigma = 0.6$ is used in our experiments. That is, slightly high weights are given to the rating score.

$$Score(c) = \sigma \times Rating(c) + (1 - \sigma) \times Length(c)$$

3.3 Experiments

To evaluate the proposed solutions, we collected the most-commented news articles and their comments from Yahoo! News and conducted experiments on this dataset. Note that our models can also be applied to the news articles with few comments. However, the aim of this research study is to summarize comments of news articles, if the news article has only a few comments, the user can easily read all the comments and the comment summarization is useless. Therefore, we only consider highly commented news articles in our study. We first evaluate the two topic models, MSTM and EXTM, using perplexity, the most widely used measure for topic model evaluation. The resultant summaries are then evaluated through a user study.

Figure 3.3: Number of comments per news article

Figure 3.4: Comment length distribution

3.3.1 Data Set

For conducting experiments, we crawled the top-100 most-commented news articles from Yahoo! News² on daily basis, from 29 April 2012 to 11 May 2012. After duplicate removal, there are 1005 unique news articles. The dataset is further processed with word stemming, removal of stopwords, and removal of words appeared fewer than 3 times in the collection.

²<http://news.yahoo.com/all-sections--most-commented/most-popular/1.html>

Figure 3.5: Comment length percentile

The total number of comments in our dataset is 1,064,655, about 1 million. Figure 3.3 plots the distribution of the number of comments per article. On average, there are 1059 comments per article. Figures 3.4 and 3.5 plot the distribution of comment length. Observe that most comments are shorter than 10 words. On the other hand, the vocabulary size of comments is 66,064. Compared to the vocabulary size of news articles 21,533, comments cover about three times of distinct words. Comments often extend topics of news articles and bring additional words to the vocabulary.

3.3.2 Evaluation

Perplexity [BNJ03] is the standard metric for evaluating topic models. It measures model ability of generalizing unseen data; lower perplexity indicates larger likelihood and better model performance. Perplexity is defined in the Equation 3.1 where D_{test} indicates the set of test documents (*i.e.*, the held-out set not used in training the model), and $p(E_c^d)$ indicates the probabilities of words in slave document c of master document d .

$$Perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{D_{test}} \sum_{c=1}^{C_d} \log p(E_c^d)}{\sum_{d=1}^{D_{test}} \sum_{c=1}^{C_d} N_{dc}}\right\} \quad (3.1)$$

In many cases, topic model is sensitive to the choices of hyper-parameters. However, through extensive evaluations, we observe that both MSTM and EXTM models are not very sensitive to the hyper-parameters on the Yahoo! News dataset. We fix the values of the hyper-parameters in the following evaluations: $\alpha^m = 0.1$, $\alpha^s = 0.1$, $\beta = 0.01$, $\gamma = 0.01$, $\eta = 0.01$, $a = 0.5$ and $b = 0.5$. The perplexities of the two models are calculated by using 10% held-out test set (100 documents) and the results reported are averaged over 3 runs.

In general, better generative performance for topic model can be achieved by setting larger number of topics. To fairly compare MSTM and EXTM, the same number of the topics were set for both models. More specifically, the number of slave topics in MSTM was the sum of the number of slave topics and number of extended topics in EXTM. Note that, in EXTM, we also set the number of slave topics same as the number of extended topics. Plotted in Figure 3.6(a), EXTM outperforms MSTM for all numbers of topics evaluated from 20 to 160. Observe that the differences in perplexities of the two models grow along the increment of number of topics. This result suggests that in EXTM, allowing words in comments to be generated from not only slave topics but also extended topics enhances the model generalization ability and guides the words of the same topic to cluster together with higher probability.

In the next set of evaluation, the effect of setting different numbers of slave topics and extended topics is evaluated. Specifically, the number of slave topics is first fixed to be 40, 50 and 60, then the number of extended topics is set be 1/4 to 4 times of slave topics. Figure 3.6(b) plots the perplexity of the experimental results. The results show that, for a fixed number of slave topics (*e.g.*, 40), the perplexity drops significantly by doubling the number of extended topics. The slope of dropping is larger when the number of extended topics is small. With the increasing number of extended topics, differences in perplexities between curves for any two fixed number of slave topics become smaller. One possible reason is that there exists an inherent number of extended topics in the documents. Therefore, setting a larger number does not benefit the model performance much.

(a) MSTM and EXTM with different number of topics. In MSTM, the number of topics equals the number of slave topics; in EXTM, number of topics is the sum of number of slave topics and extended topics.

(b) EXTM with different number of extended topics.

Figure 3.6: Perplexity of MSTM and EXTM

3.3.3 Topic Discovery

In this section, we discuss the example topics discovered by EXTM. The example topics discussed here are generated by setting both the number of master/slave topics and the number of extended topics to be 50. Table 3.2 lists 8 master topics randomly selected from the 50 topics derived from news articles. Each topic is represented by its top-ranked words in the order of their corresponding probabilities ($p(w|z^m)$). These topics suggest that EXTM discovers comprehensive and distinctive master topics. For example, Topic 10 is about presidential campaign in US, Topic 23 is about Obama’s speech on the death of Bin Laden, Topic 27 is about tax policy, and Topic 48 is related to a car accident. All these master topics are related to some hot events or popular topics during the data crawling period. This result shows that EXTM mines semantically meaningful and distinctive master topics from news articles.

The relationship between the master topics, slave topics, and extended topics is also illustrated in Table 3.2. Rows -S and -E show the most relevant slave topic and extended topic³ for the 8 master topics. Comparing master topics and the corresponding slave and extended topics, we see that EXTM model is able to capture strong relationships between the three types of topics. Master topics and their corresponding slave topics share a large number of common keywords. For instance, both master and slave topics for Topic 10 are about presidential campaign. For Topic 23 both master and slave topics are about the death of Bin Laden. Recall that the vocabulary size of slave documents is three times of the vocabulary size of master documents. Such result suggests the generative process of the slave documents is under control of that of master documents to a large extent. Next, we consider the extended topics. Extended topics are considered as extensions of master topics by news readers. Take Topic 9 as an example, both the master and slave topics are closely related to the event “Shooting of Trayvon Martin”⁴. The extended topic well captures the discussions on alleged race issues reflected from its keywords (*e.g.*, **black**,

³The topic relevance is the cosine similarity over the word-feature-vector based on the multinomial distributions of the words for topics.

⁴http://en.wikipedia.org/wiki/Shooting_of_Travon_Martin

ID	Master, Slave and Extended Topics
0-M	police protest people day immigrate illegal officer report building arrest occupy law sheriff court movement
0-S	people protest occupy ows job police time movement day america street riot idiot try property
0-E	illegal law america country mexico people joe obama border immigrate sheriff enforce government job immigrant
8-M	job time people month add april report hire times twitter launch suggest company plan look
8-S	job america people pay union country rich class money republican worker china business middle corporation
8-E	job unemployment obama people rate look month million economy america drop unemployed create count report
9-M	zimmerman martin death florida prosecutor police jones charged riot website king black people murder march university
9-S	people death kill murder martin guy zimmerman life prison police justice trial crime law america
9-E	black white people sharpston zimmerman race racist riot media martin trayvon violence jackson hate kill america racial
10-M	romney obama campaign republican president mitt candidate presidential former politics primary voter gop democratic nominee
10-S	obama romney bush america president job republican mitt vote country policy people economy gop tax war debt money
10-E	romney obama mccain mitt palin vote president endorse pick republican running sarah guy gop john
23-M	afghanistan laden bin pakistan obama attack kill america president war osama release raid house death
23-S	obama bin laden bush president kill osama credit seal romney america killing terrorist war republican
23-E	obama bin bush laden america president war afghanistan troops kill iraq osama people country romney
25-M	church smith million god mormon fox company scout international kimmel rupert catholic belief dinner faith religion religious
25-S	god jesus sin bible christ people lord love christian homosexual believe obama gods gay life
25-E	people god religion muslim christian believe bible islam world jesus church america religious mormon hate
27-M	cut rate tax billion spend debt vote student percent economy loan increase million budget business
27-S	tax pay cut rich job income obama romney rate class people bush money america spend
27-E	money tax pay spend people obama government cut america dollar country debt war military billion
48-M	police found car told time inside officer death die investigate hit injury gun son vehicle
48-S	shot fire story bullet gun hole wife kill family shoot warning attack happen die husband
48-E	road car people driver driving kid bear happen family accident time stupid sunbathe child drive

Table 3.2: Example topical keywords for master topics and their corresponding slave topics and extended topics. The master, slave, and extended topics are denoted by -M, -S, and -E respectively.

Table 3.3: Least relevant extended topics derived from comments

ID	Extended topic keywords
20	animal wild cheetah people attack pet cat pictures taking tame human stupid try maul play
11	ticket trash threw judge money lottery woman store winner found lady throw winning claim sign
15	family god bless parent little prayer heart avery love life sad story time hope pray
4	kid school parent bully child girl people children little time grade sexual teacher suspend home
6	girl scout boys play women team priest school bishop little people bar rape boy own
47	eat shrimp food kfc people chicken china baby asia chip time tiger fast america fry
24	yahoo story article comment news read people post time report reading articles picture video headline
13	movie bond character avenge love movies people film animal inmate craig james watch prison time

white, race, racist, and racial). The extended topic associated with Topic 8 is more about unemployment when talking about job issue.

Table 3.3 lists the 8 extended topics which are least relevant to master topics, in order to illustrate that some extended topics are not specific to any particular news article. Most of these 8 topics listed in Table 3.3 are casual discussions and not specific to the events reported during the crawling period. In particular, Topic 24 is about the selection of news articles and the quality of services of Yahoo! News in general. Topic 15 is about readers' sentiment on news stories.

3.4 User Study

We now evaluate the summaries generated by the proposed methods through a user study. Document summarization is usually evaluated by comparing with ground truth summaries (*e.g.*, summaries generated by human being or manually-labeled sentences). In our problem setting, generating such ground-truth labeling is extremely difficult because the number of comments to each news article is a large number (*i.e.*,

1059 comments per news article on average). Moreover, *topics* are not predefined. As the result, it is hard to manually label the comments to form the ground truth. In our user study, we therefore directly evaluate the summaries generated by the different methods.

Methods. We evaluated 6 methods in total. The 6 methods involve 3 clustering methods CLUTO, MSTM and EXTM, and 2 ranking methods RL and MMR. For CLUTO⁵, we used bisecting clustering algorithm (with default parameter settings) as a method to cluster comments without using the notion of topic. CLUTO was chosen to cluster comments because of its high efficiency and promising performance in many reported clustering tasks. The number of clusters was set to 50. Note that bisecting clustering was applied on the set of slave documents of each master document independently. Based on the clusters generated by CLUTO, the comment summaries selected by MMR and RL respectively are denoted by CLUTO-MMR and CLUTO-RL. In CLUTO-MMR, $TF \times IDF$ based word weighting scheme and cosine similarity were used.

For the two topic models MSTM and EXTM, the comments are clustered based on the topics discovered by the models. The number of topics for MSTM was set to 100. The number of slave topics and the number of extended topics were both set to 50 in EXTM. Postfixes -MMR and -RL are used to distinguish the MMR and RL comment selection method.

Evaluation Setting and Criteria. For user evaluation, 50 articles from the 1005 news articles are randomly selected. For each selected article, the number of comments selected in a comment summary is set to 15, covering 3 major topics discussed in comments. More specifically, given all comments of a news article, a summarization method first selects 3 largest clusters (based on the number of comments) to represent the 3 major topics discussed in comments. From each cluster, 5 most representative comments are selected. In short, a comment summary for a news article contains

⁵<http://glaros.dtc.umn.edu/gkhome/views/cluto>

Figure 3.7: User Evaluation

exactly 15 comments covering 3 major topics. A volunteer therefore needs to read $50 \times 15 \times 6 = 4500$ comments in total, to evaluate all the 6 methods.

For each comment summary to be evaluated, a volunteer gives a rating from 1 (lowest) to 5 (highest) with respect to the following three criteria.

- *Topic cohesion*: To what extent the 5 comments in each comment cluster are topically cohesive.
- *Topic diversity*: To what extent the 3 topic clusters cover 3 different topics.
- *News relatedness*: To what extent the 3 topic clusters are related to the content of the news article.

Evaluation Results. Figure 3.7 plots the averaged ratings from three volunteers (undergraduate students with computer science background) by the method producing the comment summary. In addition to the three criteria, the average rating of the three criteria is also plotted in the last column. We make three observations.

First, across all the three clustering methods, MMR ranking outperforms RL ranking. A possible reason is that RL suffers from rating sparsity and its resultant ranking by length is biased to long (and maybe even redundant) comments. Both hurt topic cohesion and topic diversity. Conversely, MMR tries to select comments that

are closest to the centroid of a comment cluster but with different word distributions because of its redundancy control. In particular, EXTM-MMR yields the best topic cohesion rating of 3.59 and the best topic diversity rating of 3.52. It outperforms CLUTO-MMR by 14% on topic cohesion and by 13% on topic diversity, respectively.

Second, comparing with traditional clustering approach CLUTO, topic model based methods are more likely to generate clusters with better intra-cohesive and inter-diverse measures. However, CLUTO methods achieve the best news relatedness. The reason is that CLUTO is independently applied on all the comments of a single news article. The clustering is therefore not affected by any additional (or contextual) information reported in other news articles or comments. Both MSTM and EXTM based methods naturally capture such contextual information. However, such contextual information may not be known to our volunteers due to the random selection of the news articles used in this evaluation. For instance, when clustering the comments of a news article talking about president election in US, we observe that most comments are about the president election and some comments are about unemployment in US. The top 3 topical comment clusters generated by CLUTO merely focus on who will be the next president in US, because only comments of this news article are considered by CLUTO and the only dominant topic on president election related to the news article is discovered. However, when applying MSTM and EXTM to the comments of this news article, the effect of other news articles about unemployment is also introduced, which increases the weight of topical comment cluster about unemployment. Therefore, the top 3 topical clusters generated by MSTM and EXTM include the cluster about unemployment, which causes the decrease of news relatedness. Topic model based methods are rated lower on the measure of news relatedness.

Third, between MSTM and EXTM, EXTM-based methods outperform MSTM-based methods on topic cohesion and topic diversity, but are slightly poorer on news relatedness. The lower news relatedness rating may have the reason that EXTM discovers extended topics discussed solely among comments, but not in news articles. With all

evaluation criteria of equal importance (*i.e.*, the average of the three), EXTM-MMR is the best performing method among all the 6 methods evaluated.

3.5 Summary

In this chapter, we first define the topic-driven reader comment summarization problem (also called comment ranking for news articles problem). The main objective of the research is to present to users a summary of comments covering the major topics discussed among the readers. After analyzing the important relationships between news and comments, we propose two generative models, namely, Master-Slave Topic Model and Extended Master-Slave Topic Model. Both models implicitly model the identified relationships between news article and comments. Based on the master-slave structure, both models treat news article as master document and comment slave document. Because of the shortness of comments, both models assume that each comment has exactly one topic. While MSTM constrains that the topic of a slave document has to be derived from its master document, EXTM allows the topic of a slave document derived from all slave documents, which allows the generation of extended topics that are discussed by users but not covered in the news articles. Evaluated using perplexity, in our experiments EXTM significantly outperforms MSTM for grouping comments with the same topics. To generate the comment summary, we utilize two comment ranking schemes MMR and RL to select the most representative comments from each comment cluster. Our user study shows that EXTM-MMR outperforms the other 5 methods by topic-cohesion and topic-diversity measures.

Chapter 4

Hashtag Ranking for Tweet Annotation

In this chapter, in order to address the problem of hashtag ranking for tweet annotation, we study two important factors, tweet content (master document) and hashtag (slave document), and their relationships. We also evaluate two additional factors (*i.e.*, user and time) that affect the adoption of hashtags. Two PLSA-style topic models are proposed to model the hashtag annotation behavior in Twitter. ***Content-Pivoted Model*** (CPM) assumes that tweet content guides the generation of hashtags while ***Hashtag-Pivoted Model*** (HPM) assumes that hashtags guide the generation of tweet content. Both models jointly incorporate user, time, hashtag, and tweet content in a probabilistic framework. The PLSA-style models also enable us to verify the impact of social factor on hashtag annotation by introducing social network regularization in the two models. Our results show that HPM outperforms CPM by perplexity and both user and time are important factors that affect model performance. In addition, incorporating social network regularization does not improve model performance. Our experimental results also demonstrate the effectiveness of our models on the problem of hashtag ranking for tweet annotation compared with baseline methods.

The rest of this chapter is organized as follows. Section 4.1 briefly review the related work. Section 4.2 present our two models, Content-Pivoted Model and Hashtag-Pivoted Model and their inference algorithms. In addition, we also show our models with social network regularization in Section 4.2. Section 4.3 evaluates the proposed models measured by perplexity and discovered topics. Section 4.4 applies our models to the problem of hashtag ranking for tweet annotation and related hashtag discovery. Section 4.5 summarizes this chapter.

4.1 Related Work

Hashtags in Twitter. Hashtags have attracted much research attention because of their wide adoption. Many different perspectives of hashtags have been studied in the literature, such as hashtag popularity prediction [TR12, KC13], hashtag adoption prediction [YSZM12], hashtag diffusion [RMK11], hashtag sentiment analysis [WWL⁺11, DTR10], hashtag for event detection [CWCC14] and hashtag for group chatting [CKM13].

In [TR12], Tsur and Rappoport used regression model to predict hashtag popularity on weekly basis. Both features derived from tweet content and features from hashtag itself (*e.g.*, orthography, number of characters in a hashtag) are used in the prediction. Their experimental results demonstrate that content features improve prediction performance. For the issue of hashtag adoption prediction [YSZM12], Yang *et al.* discovered that there are two main goals for hashtag adoption, joining a community on the same topic or trend and bookmarking tweet content. The features include (i) *prestige* and *influence* derived from social graph formed by users who adopt a hashtag, and (ii) *relevance* and *preference* derived from tweets with hashtags. A strong relationship between hashtag adoption and tweet content has been revealed in both previous studies. In our study, for capturing the relations between hashtag and tweet content, we jointly model the two factors at the topic level.

In [RMK11], hashtags have been categorized into 8 classes (*e.g.*, *politics*, *celebrity*, and *game*) and the differences in the mechanics of information diffusion of hashtags from different classes have been analyzed. The authors showed that the adoption time of hashtags on *politics* is longer than that of hashtags from other classes. Moreover, the exposure time of a hashtag (*i.e.*, how many times a user observes this hashtag in her Twitter stream) plays a crucial role in hashtag diffusion. In order to trace information diffusion, the authors constructed a user graph based on mention relationship. The impact of user factor on hashtag adoption is also considered in our work and the social network is treated as a regularization in our model, assuming that users who often mention each other are more likely to share common topics.

Hashtag Recommendation. Tag recommendation is an important research topic in recent years. Two main techniques (*i.e.*, tensor factorization [RST10, SNM08, RBMNST09] and graph model [GBM⁺09, FW12]) have been proposed and applied to social tagging systems like Delicious and Flickr. For hashtag recommendation [ZGS11, MJ, KHLZ12, SS14] in Twitter, both tweet-based recommendation and user-based recommendation [DADSTN12, LXTC12] have been proposed. Next, we briefly review tweet-based recommendation which is more relevant to our work.

To recommend a hashtag list for given tweet, in [ZGS11], the authors searched for similar tweets to the given tweet by computing content similarity, then ranked the hashtags in these similar tweets by their usage. Mazzia *et al.* [MJ] also employed a Bayesian model and used tweet content for hashtag recommendation. Furthermore, Kywe *et al.* [KHLZ12] incorporated user preference into the model [ZGS11]. That is, hashtags to be recommended to a tweet d by user u are the hashtags used to annotated many similar tweets to d and the hashtags adopted by many similar users to u . In our study, we use this approach to be our baseline approach for the problem of hashtag ranking for tweet annotation.

Table 4.1: Notations for CPM and HPM

Symbol	Description
D	Collection of tweets
H	Collection of hashtags
U	Collection of users
T	Collection of time slots
D_u	Collection of tweets posted by user u
D_t	Collection of tweets posted at time slot t
D_h	Collection of tweets having hashtag h
D_w	Collection of tweets having word w
$\mathbf{w}_d$	Collection of words in tweet d
$\mathbf{h}_d$	Set of hashtags in tweet d

4.2 Hashtag Annotation Models

In this section, we present the two hashtag annotation models: *Content-Pivoted Model* (CPM) and *Hashtag-Pivoted Model* (HPM). Both models jointly model tweet content, user, time, and hashtag, but with different assumptions on the generation of hashtag and tweet content. In the following, we start with the notations used in our models and the intuitions in our models. We then present the two models and their inference algorithms. Lastly, we detail the inference algorithms considering social network regularization in the two models CPM and HPM. The models with social network regularization are denoted by CPM^{sn} and HPM^{sn} respectively.

Notations. Let d be a tweet and D be a collection of tweets. Let U be a collection of users each of which has published at least one tweet. We partition time into a sequence of time slots of fixed length and map the publication time of a tweet to a time slot t .¹ Let T be the collection of time slots, V be the word vocabulary, and E be the hashtag vocabulary. A tweet d is a 4-tuple $d = \{u, t, \mathbf{w}_d, \mathbf{h}_d\}$: $u \in U$ is the author of the tweet; $t \in T$ is the time slot within which d was published; $\mathbf{w}_d$ is the word collection in d , where the words are drawn from V ; and $\mathbf{h}_d$ is the set of distinct hashtags annotated to tweet d , where the hashtags are drawn from E . Note that, a

¹In this chapter, the length of a time slot is a day.

tweet may have more than one hashtag and even duplicated hashtags. In our work, we only consider unique hashtags for the same tweet.

Intuitions and Assumptions. All our models are designed based on the following two intuitions:

- The topic of a tweet is guided by the personal interest (or activity) of the user who has published this tweet. A large portion of tweets reflect the users' personal interests or activities. Based on this intuition, we model each user as a topic probability vector. The topic of a tweet from a user is generated based on her corresponding topic probability distribution.
- The topic of a tweet is also guided by time. A large number of tweets have strong relations with recent and ongoing events or trending topics. More specifically, each time slot is associated with some major events or popular topics happened or discussed within that time slot. Tweets published in different time slots reflect different topic distributions. Based on this intuition, each time slot is modeled as a topic probability vector and the topic of a tweet published in that time slot could be sampled based on its corresponding topic probability.

A hashtag is a high-level abstraction of the tweet content. Among all factors, words in a tweet is the most import factor affecting hashtag annotation. However, when composing a tweet with hashtag(s), there could be two possible cases: (i) user composes the tweet first and then finds appropriate hashtags to annotate this tweet, or (ii) user has a hashtag (*e.g.*, a hashtag created for a popular event) in mind and writes a tweet for the hashtag. To model the difference in the order of generating tweet content and hashtags, we propose two models: Content-Pivoted Model which assumes the tweet content is drafted first and the generation (or selection) of the hashtag is guided by the tweet content, and (ii) Hashtag-Pivoted Model which assumes that the user has selected the hashtag and then drafts the tweet content based on her understanding of this hashtag. In both models, we assume that each tweet has only one topic due to its short length. The same assumption has been adopted in many

Figure 4.1: Graphical model representations of *CPM* (without dotted line) and *CPM^{sn}* (with dotted line).

other studies [DJZL12,ZJW⁺11]. In the following, we detail the two models and their inference algorithms.

4.2.1 Content-Pivoted Model (CPM)

Figure 4.1 (without dotted line) illustrates the Bayesian graphical representation of CPM. Some of the notations used in the model are summarized in Table 4.1.

The topic z of tweet d is generated from personal interest of u or topic distribution of time slot t . That is, when user u publishes a tweet d in time slot t , she first decides whether to write anything related to her personal interests/activities or to comment on some hot topics in that time slot. More specifically, the topic z of tweet d can be generated from the user topic distribution $p(z|u)$ and the time topic distribution $p(z|t)$. We use a parameter α to balance the importance between $p(z|u)$ and $p(z|t)$:

$$p(z|u, t) = \alpha p(z|u) + (1 - \alpha) p(z|t)$$

After a topic z is generated, all words $\mathbf{w}_d$ in the tweet d are sampled from $p(w|z)$. Then the hashtags of tweet d , $\mathbf{h}_d$, are sampled from $p(h|z)$. The generative process of CPM is summarized as follows:

- For each tweet $d \in D$, written by user u at time t
 - Draw a topic $z \sim p(z|u, t)$
 - For each word w in $\mathbf{w}_d$, draw $w \sim p(w|z)$
 - For each hashtag h in $\mathbf{h}_d$, draw $h \sim p(h|z)$

Note that, in CPM model, all factors (*i.e.*, user, time, tweet content, and hashtag) are incorporated into a PLSA framework. Furthermore, a topic has been first sampled based on the user interests or (popular) topics of that time slot, and then the words of tweet content and hashtags are generated from the topic.

4.2.2 Hashtag-Pivoted Model (HPM)

Similar to CPM, the HPM model also jointly considers user, time, tweet content and hashtag. However, as illustrated in Figure 4.2 (without dotted line), HPM models hashtags as a high-level feature partially guiding the generation of tweet content. That is, when composing a tweet, a user may choose to post her personal interests or comment on some hot events in that time slot as in CPM; while in HPM a user may also choose to directly comment on a specific hashtag. Formally, the topic z of a tweet may be drawn from user topic distribution $p(z|u)$, time topic distribution $p(z|t)$, or hashtag topic distribution $p(z|\mathbf{h}_d)$. Observe that, one tweet might have more than one hashtags. We assume that all hashtags of a tweet $\mathbf{h}_d$ share equal importance to the tweet d :

$$p(z|\mathbf{h}_d) = \frac{1}{|\mathbf{h}_d|} \sum_{h' \in \mathbf{h}_d} p(z|h')$$

Considering the three factors $p(z|u)$, $p(z|t)$, $p(z|\mathbf{h}_d)$, the topic z of a tweet d written by a user u at time slot t with hashtag(s) $\mathbf{h}_d$ in mind is:

Figure 4.2: Graphical model representations of HPM (without dotted line) and HPM^{sn} (with dotted line).

$$p(z|u, t, \mathbf{h}_d) = \beta(\alpha p(z|u) + (1 - \alpha)p(z|t)) + (1 - \beta)p(z|\mathbf{h}_d) \quad (4.1)$$

Here, we use the two parameters α and β to balance the importance of the three factors in selecting the topic of the tweet. Similarly, after generating the topic z of tweet d , all words $\mathbf{w}_d$ are sampled from $p(w|z)$. The generative process of HPM is as follows:

- For each tweet $d \in D$, written by user u at time t for hashtags $\mathbf{h}_d$
 - Draw a topic $z \sim p(z|u, t, \mathbf{h}_d)$
 - For each word w in $\mathbf{w}_d$, draw $w \sim p(w|z)$

4.2.3 Inference Algorithms

The latent variable topic z is required to be inferred for both CPM and HPM. The exact inference algorithm is intractable. An Expectation-Maximization (EM) algorithm for appropriately inferring z in both models is employed in our study. Next, we first detail the inference algorithm for CPM. In CPM, the joint probability over tweet d and topic z can be represented as:

$$\begin{aligned} p(d, z) &= p(u, t, z, \mathbf{w}_d, \mathbf{h}_d) \\ &= p(u)p(t)p(z|u, t)p(\mathbf{w}_d|z)p(\mathbf{h}_d|z) \end{aligned} \quad (4.2)$$

where

$$p(z|u, t) = \alpha p(z|u) + (1 - \alpha)p(z|t) \quad (4.3)$$

$$p(\mathbf{w}_d|z) = \prod_{w' \in \mathbf{w}_d} p(w'|z) \quad (4.4)$$

$$p(\mathbf{h}_d|z) = \prod_{h' \in \mathbf{h}_d} p(h'|z) \quad (4.5)$$

Accordingly, the log-likelihood in CPM is $L = \sum_d \log \sum_z p(d, z)$. We train the model using EM algorithm as follows:

- In **E-step**,

$$p(z|d) = \frac{p(d, z)}{p(d)} = \frac{p(d, z)}{\sum_z p(d, z)} \quad (4.6)$$

- In **M-step**, it is complicated to estimate $p(z|u)$ and $p(z|t)$, because they are coupled by the sum in logarithm in log-likelihood, *i.e.*, $\log(\alpha p(z|u) + (1 - \alpha)p(z|t))$. We apply Jensen's inequality to get a lower bound: $\log(\alpha p(z|u) + (1 - \alpha)p(z|t)) \geq \alpha \log p(z|u) + (1 - \alpha) \log p(z|t)$. We now maximize the log-likelihood to estimate

the following parameters:

$$p(z|u) = \frac{\sum_{d \in D_u} p(z|d)}{\sum_{d \in D_u} \sum_{z'} p(z'|d)} \quad (4.7)$$

$$p(z|t) = \frac{\sum_{d \in D_t} p(z|d)}{\sum_{d \in D_t} \sum_{z'} p(z'|d)} \quad (4.8)$$

$$p(w|z) = \frac{\sum_{d \in D_w} n(d, w) p(z|d)}{\sum_{w'} \sum_{d \in D_{w'}} n(d, w') p(z|d)} \quad (4.9)$$

$$p(h|z) = \frac{\sum_{d \in D_h} p(z|d)}{\sum_{h'} \sum_{d \in D_{h'}} p(z|d)} \quad (4.10)$$

where $n(d, w)$ represents number of appearances of word w in d , or w 's term frequency in d .

The joint probability for HPM over tweet d and topic z is defined in the following equation, where $p(z|u, t, \mathbf{h}_d)$ is defined in Equation 4.1:

$$\begin{aligned} p(d, z) &= p(u, t, \mathbf{h}_d, z, \mathbf{w}_d) \\ &= p(u)p(t)p(\mathbf{h}_d)p(z|u, t, \mathbf{h}_d)p(\mathbf{w}_d|z) \end{aligned} \quad (4.11)$$

In Equation 4.11, $p(\mathbf{h}_d) = \prod_{h' \in \mathbf{h}_d} p(h')$. The inference algorithm for HPM is similar to that of CPM. Specifically, the E-steps for both models are the same. In M-step, the estimations of $p(z|u)$, $p(z|t)$, and $p(w|z)$ in CPM also apply to HPM. The additional parameter $p(z|h)$ in HPM is estimated as follows:

$$p(z|h) = \frac{\sum_{d \in D_h} p(z|d)/|\mathbf{h}_d|}{\sum_{d \in D_h} \sum_{z'} p(z'|d)/|\mathbf{h}_d|} \quad (4.12)$$

4.2.4 Complexity Analysis

We now analyze the time complexity of our proposed CPM and HPM. For CPM, in the E-step, the time complexity is $O(MK|D|)$, where M , K , $|D|$ are the number of iterations, the number of topics and the number of tweets, respectively. In the M-step,

the time complexity is $O(MK(|U||D_u^{avg}| + |T||D_t^{avg}| + |W||D_w^{avg}| + |H||D_h^{avg}|))$, where $|U|$, $|T|$, $|W|$, $|H|$ are the number of users, the number of time slots, the vocabulary size of words and the vocabulary size of hashtags, respectively. $|D_u^{avg}|$, $|D_t^{avg}|$, $|D_w^{avg}|$ and $|D_h^{avg}|$ indicate the average number of tweets written by user u , the average number of tweets posted in time slot t , the average number of tweets containing word w , and the average number of tweets containing hashtag h . The space complexity depends on the probabilities required to be stored, and it is $O(K(|D| + |U| + |T| + |W| + |H|))$. The time and space complexity for HPM are the same as that in CPM.

4.2.5 Social Network Regularization

It is possible that more similar topics are shared by users who often mention each other. In order to obtain more accurate topics from the Twitter data, the mention relationship in Twitter is introduced into our models. This relationship is considered as a regularization R over the topic distribution of a pair of Twitter users who have mentioned each other. Formally, we minimize the proximity of topic distributions $p(z|u)$ and $p(z|v)$ of two users u and v who have mentioned each other for C_{uv} number of times in their tweets (regardless u mentions v or v mentions u):

$$R = \sum_{u,v \in U} \sum_z C_{uv} (p(z|u) - p(z|v))^2$$

CPM^{sn} and HPM^{sn} denote CPM and HPM with social network regularization respectively. The dotted lines in Figures 4.1 and 4.2 denote the social network regularization. Regularized log-likelihood is calculated as $RL = L - \lambda R$, where λ is the regularization parameter. Note that we set $\lambda = 10$ in our evaluation following the setting in [GCXZ11]. We maximize the regularized log-likelihood using Generalized EM algorithm [NH99].

Except for $p(z|u)$, all other parameters in CPM^{sn} and HPM^{sn} are estimated in the same way as their corresponding models CPM and HPM. Next, we use CPM^{sn} as an example to estimate $p(z|u)$ and the same applies to HPM^{sn} . Let $p^i(z|u)$ be the

estimation obtained in the i -th iteration of CPM^{sn} , $p^{i+1}(z|u)$ in the $(i+1)$ -th iteration is computed using Equation 4.13 based on the Newton-Raphson method [PFTV88]. Note that $p^0(z|u)$ is the $p(z|u)$ estimated in CPM (see Equation 4.7).

$$p^{i+1}(z|u) = (1 - \gamma)p^i(z|u) + \gamma \frac{\sum_{v \in U} C_{uv} p^i(z|v)}{\sum_{v \in U} C_{uv}} \quad (4.13)$$

In the above equation, γ is the step parameter ($\gamma = 0.1$ in our implementation following the setting in [GCXZ11]) and C_{uv} is the number of times users u and v who have mentioned each other in their tweets. More details of the algorithm can be found in [GCXZ11].

4.3 Experiments

We conducted experiments on Twitter data set and the performance of CPM and HPM is evaluated by perplexity. We also illustrated example topics discovered by our models. Moreover, we evaluated the impact of social network regularization in CPM and HPM.

4.3.1 Data Set

The tweets used in our evaluation are published by Singapore-based users from 1 January 2011 to 31 August 2011.² Because our study works on the hashtag annotation behavior, tweets without hashtags are filtered out in our evaluation. In other words, in our experiments, each tweet contains at least one hashtag. We also removed stopwords and non-English words in all tweets and then dropped tweets with empty content. We removed tweets annotated with extremely infrequent hashtags (*i.e.*, each is used to annotate fewer than 5 tweets in the whole collection) from our collection to ensure that each hashtag has a reasonable number of tweets for topic modeling. As the result,

²User location information is based on the location specified in user profile.

(a) Hashtag frequency distribution

(b) Number of hashtags per tweet

Figure 4.3: Hashtag frequency distribution and number of hashtags per tweet

Table 4.2: Statistics of the data set

Number of tweets $ D $	1,217,928
Number of distinct hashtags $ E $	14,055
Number of distinct words or vocabulary size $ V $	61,274
Number of users $ U $	13,711
Number of time slots (days) $ T $	243

every hashtag in our final collection has been used to annotate at least 5 non-empty tweets written in English.

After preprocessing, our experimental data set contains more than 1.2 million tweets published by over 13 thousand users in 243 days. More than 14 thousand distinct hashtags are used to annotate the tweet collection. The statistics of our processed data set are reported in Table 4.2.

The hashtag frequency distribution follows a power-law like distribution (see Figure 4.3(a)), which indicates that a small number of hashtags are extremely popular to annotate many tweets, while most hashtags are used few times by few users. Note that 82.2% of tweets in our collection has one hashtag each (see Figure 4.3(b)). The remaining 17.8% of tweets, each has more than one hashtag. The number of tweets with more than 10 hashtags each is very small.

4.3.2 Evaluation by Perplexity

The standard metric for evaluating topic models [BNJ03] is *perplexity*. Defined in Equation 4.14, perplexity measures the ability of a model in generating unseen data (*i.e.*, D_{test} in the equation, which is a document collection for testing). In this equation, $p(\mathbf{w}_d)$ indicates the probability of generating all the words in a test document $d \in D_{test}$, and N_d denotes the number of words in document d . Lower perplexity indicates better model performance.

$$Perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d \in D_{test}} \log p(\mathbf{w}_d)}{\sum_{d \in D_{test}} N_d}\right\} \quad (4.14)$$

 (a) *CPM*

 (b) *HPM* ($\alpha=0.9$)

 (c) *HPM* ($\beta=0.6$)

 Figure 4.4: Perplexity of *CPM* and *HPM* with varying α , β or K .

Figure 4.5: The impact of social network regularization on *CPM* and *HPM* ($K=100$)

In our experiments, 200,000 tweets are randomly selected as the testing data set, and the remaining 1,017,928 tweets are used as training data set. Next, we first examine the impact of user factor, time factor and the number of topics on the model performance of *CPM* and *HPM* respectively by perplexity. We then evaluate the effectiveness of social network regularization on the two models by comparing their perplexity with that of CPM^{sn} and HPM^{sn} . In all our experiments, the number of iteration in training the models is fixed to 100.

CPM Model Performance. Recall that in *CPM*, the topic z of a tweet d is generated from the user topic distribution $p(z|u)$ and the time topic distribution $p(z|t)$, balanced with a parameter α : $p(z|u, t) = \alpha p(z|u) + (1 - \alpha)p(z|t)$. To evaluate the impact of user interest $p(z|u)$ and time factor $p(z|t)$, we vary α from 0 to 1, with a step of 0.1. Observe that when $\alpha = 0$, topic z is generated from time topic distribution only; and when $\alpha = 1$, topic z is generated purely based on user interest. Figure 4.4(a) plots the perplexity of *CPM* with varying α from 0 to 1 for four topic number settings $K = \{25, 50, 100, 150\}$. We make three observations from this result.

First, parameter α has a significant impact on the perplexity of the model which is in the range from 3800 to 5400. With the four different topic number settings, the

perplexity values follow very similar trends against the varying of α . When $\alpha = 0.9$, the lowest perplexity is achieved for all the four topic number settings. Either $\alpha = 0$ or $\alpha = 1$ results in much poorer model performance, indicating that (i) both user interest and time are important factors affecting the topic of tweets, and (ii) user interest is often the dominant factor in determining the topics of the tweets from a user.

Second, regarding the choice of number of topics, $K = 25$ or $K = 150$ leads to poorer performance than $K = 50$ or $K = 100$. Particularly, $K = 100$ and $\alpha = 0.9$ delivers the best perplexity in this set of experiments. In all our following experiments, we therefore set $K = 100$ and $\alpha = 0.9$ as the default settings.

Third, when tweet topics are purely drawn from time topic distributions (*i.e.*, $\alpha = 0$), the number of topics K has a limited impact on the perplexity. However, when tweet topics are solely generated based on user interest (*i.e.*, $\alpha = 1$), the smaller the number of topics (*i.e.*, $K = 50$), the better the perplexity. This observation suggests that a common user usually does not show interests in too many different topics.

HPM Model Performance. Compared with CPM, HPM considers one more factor $p(z|\mathbf{h}_d)$ in generating the topic of a tweet. More specifically, $p(z|u, t, \mathbf{h}_d) = \beta(\alpha p(z|u) + (1 - \alpha)p(z|t)) + (1 - \beta)p(z|\mathbf{h}_d)$. Note that $\beta = 0$ leads to tweet topic generation solely based on hashtags $p(z|\mathbf{h}_d)$.

Based on the results of CPM, we first set $\alpha = 0.9$ and evaluate the perplexity of HPM against the varying of β from 0 to 1 with a step of 0.1. Demonstrated in Figure 4.4(b) the impact of β on HPM is not significant when $\beta \geq 0.1$ for all K values. When $K = 100$, HPM achieves the best perplexity when $\beta = 0.6$. However, it is observed that the perplexity is much poorer when $\beta = 0$, *i.e.*, the topic of a tweet is purely generated based on hashtags.

Next, we fix $\beta = 0.6$ and vary the values of α from 0 to 1 (see Figure 4.4(c)). Similar to that in CPM, the perplexity of HPM is best when $\alpha = 0.9$ for all the four different numbers of topics. Compared with CPM, HPM performs better by perplexity,

with perplexity ranging from 3600 to 4100. One reason is that HPM treats hashtags as topic vectors which could better cluster the words in tweets leading to better topic cohesion.

Social Network Regularization. We now evaluate the impact of considering social factor in the two models. In this set of experiments, we set number of topics $K = 100$ for all four models: CPM, CPM^{sn} , HPM and HPM^{sn} . For both HPM and HPM^{sn} , β is set to 0.6 based on earlier experimental results. The two additional parameters γ and λ in CPM^{sn} and HPM^{sn} are experimentally set to $\gamma = 0.1$ and $\lambda = 10$ (see Section 4.2.5).

Figure 4.5 shows the perplexity of all four models with α varying from 0 to 1. User interest is excluded when $\alpha = 0$, therefore the model ignores social factor at that point. As shown in Figure 4.5, the performance (in term of perplexity) of both models is much worse when introducing the social network regularization. One possible reason is that, two users may mention each other because of common interests in some but not all the topics. The assumption that a pair of users who mention more about each other are more likely to share similar topic distributions might be too strong. However, on the other hand, predetermining a subset of common topics for a given pair of users is infeasible in generative models.

4.3.3 Topic Discovery

We now present 7 sample topics discovered by the two models CPM and HPM. For both models, we set the number of topics to be 100. From the 100 topics, 7 topics are selected as examples. These 7 topics are listed in Table 4.3. To better explain them, these 7 topics are manually labeled. Because word probability $p(w|z)$ and hashtag probability $p(h|z)$ (see Section 4.2.1) can be generated from the CPM model, we show both the top words and the top hashtags of the 7 sample topics based on their generative probabilities. For clarity, we name these two kinds of topics *word topic* and *hashtag topic* respectively. For HPM, only word topic based on $p(w|z)$ (see Sec-

Topic label	Topic type	Top-10 words/hashtags with highest generative probability
job	<i>CPM</i> Hashtag	#job #jobs #career #interview #hr #jobhunt #jobsearch #sg #interviews #re-cruit
	<i>CPM</i> Word	questions interviewer job difference photo seeking using hour building rapport
	<i>HPM</i> Word	job singapore basic based industry questions executive days interviewer sales
singapore election	<i>CPM</i> Hashtag	#sgpresident #sgelections #singapore #sgelection #sgpolitics #fb #news #cars2race #yamahmee #pe2011
	<i>CPM</i> Word	tan tony president cheng dr jee bock kin vote lian
	<i>HPM</i> Word	tan tony cheng dr jee president bock kin presidential lian
japan earthquake	<i>CPM</i> Hashtag	#prayforjapan #japanlife #fb #japan #tsunami #sgelections #singapore #pray-fortheworld #earthquake #oscars
	<i>CPM</i> Word	japan life live earthquake hope join people goal please tsunami
	<i>HPM</i> Word	japan god please hope earthquake singapore people safe news tsunami
digital devices	<i>CPM</i> Hashtag	#technews #technology #singapore #apple #fb #socialmedia #google #news #simonvideo #jobs
	<i>CPM</i> Word	apple iphone ipad video google app social facebook android media
	<i>HPM</i> Word	apple iphone ipad app google android video mobile phone mac
music	<i>CPM</i> Hashtag	#nowplaying #replacesongnameswithcurry #replacesongnameswithbangla #np #replacebandnameswithbangla #singapore #fb #nowlistening #lastfm #things-brokepeopleledo
	<i>CPM</i> Word	curry love perry katy bangla adele rock black rolling party
	<i>HPM</i> Word	curry love song listening mars 987 bruno perry lt3 katy
football	<i>CPM</i> Hashtag	#lfc #mufc #fb #manutd #ynwa #arsenal #sleague #singapore #ff #sgfoot-ball
	<i>CPM</i> Word	play game win time match singapore friends united goal liverpool
	<i>HPM</i> Word	united win game cant team fans cup liverpool manchester arsenal
daily life	<i>CPM</i> Hashtag	#100factsaboutme #fml #fb #likeaboss #fail #nowplaying #justsaying #foreveralone #sosingaporean #random
	<i>CPM</i> Word	school lol time day sleep cant haha people gonna home
	<i>HPM</i> Word	school day time tomorrow homework study week gonna days doing

Table 4.3: Example topics with *CPM* topical words, *CPM* topical hashtags and *HPM* topical words

Table 4.4: Example topics found by *HPM* but not *CPM*

Topic label	Words with highest generative probability
royal wedding	wedding kate club quay clarke royal river demi rd valley
food	singapore food news paying restaurant world bill cash hotel free
business	ltd pte singapore manager executive sales assistant services tfeeds jeffs
shopping	parade tampines st blk 33 corner 314 gaming sunnys marine

Figure 4.6: Top-10 topics to each hashtag ranked by probability $p(z|h)$ in descending order

tion 4.2.2) is generated. As reported in [NBG11], Twitter topics can be classified into *endogenous topics* and *exogenous topics*. Endogenous topics (*e.g.*, #10thingsihate and #nowplaying) are originated within Twitter and exogenous topics (*e.g.*, #earthquake and #flood) are originated outside of Twitter. Observe in Table 4.3, the major topics discussed by Singapore users in Twitter from January to August 2011 are captured by both CPM and HPM models. Among them *Jobs*, *music* and *daily life* are example continuous endogenous topics discussed in Twitter and *Singapore General Election*³ and *Japan earthquake*⁴ are major exogenous events in our data set. Generally, both exogenous topics and endogenous topics can be explored by our models.

³http://en.wikipedia.org/wiki/Singaporean_general_election,_2011

⁴http://en.wikipedia.org/wiki/2011_Tohoku_earthquake_and_tsunami

Table 4.5: Topical keywords of the top-3 topics for each hashtag

Hashtag	$p(z h)$	Top-5 Keywords
#sgelections	0.209	pap vote grc aljunied wp
	0.163	pap rally grc tan wp
	0.110	lee pap singapore pm minister
#royalwedding	0.223	wedding kate club quay clarke
	0.065	sleep time cant gonna watch
	0.060	trending lol omg lt3 happy
#prayforjapan	0.467	japan god please hope earthquake
	0.050	love people life youre time
	0.025	news tan reuters singapore japan

Next we discuss the relationship between word topic and hashtag topic generated by CPM (see rows labeled by “CPM Hashtag” and “CPM Word” in Table 4.3). Observe that most top-ranked hashtags of hashtag topic and the corresponding word topics have strong semantic relations. Take the first topic **job** as an example, the top-ranked hashtags (*i.e.*, #**job**, #**jobs**, #**career**, #**interview**) and the top-ranked words (*i.e.*, questions, interviewer, job, difference) are closely associated semantically. Generally speaking, topical words of CPM are relatively more specific while topical hashtags of CPM are more general. However, because a topic is usually annotated by few dominant hashtags only, the top-10 hashtags listed for each topic in Table 4.3 might not all describe the corresponding topic. For instance, hashtag #**oscars** is not very relevant to **Japan earthquake** and #**royalwedding** is irrelevant to **Harry Porter movie**. Some hashtags are extremely popular (*e.g.*, #**fb**, #**singapore**) and are often used to annotate many different topics.

Only word topics (see rows labeled by “HPM Word” in Table 4.3) are generated from HPM. Some of the top-ranked topical words of CPM and HPM are very similar. The topic labeled **digital devices** is an example. However, several topics (see Table 4.4) can be found by only HPM but not CPM. The example topics are **business**, **royal wedding**, **food**, and **shopping**. HPM has stronger ability for finding less popular topics like **shopping** and **business**, which also partially explains why the perplexity of HPM is better than that of CPM.

Next, we demonstrate the topic distribution of three example hashtags generated by HPM. The three example hashtags are `#royalwedding`, `#sgelections` and `#prayforjapan`. Top-10 topics of each example hashtags ranked by probability $p(z|h)$ in descending order are listed in Figure 4.6. Note that both `#royalwedding` and `#prayforjapan` were popular for about two weeks, a relatively short time period. Each of the two hashtags has one dominant topic with the highest probability. For example, the probability of the top topic for `#prayforjapan` is nearly 50%. For `#sgelections`, it was popular for a few months and was adopted to annotate tweets for two elections (parliamentary general election and presidential election). Three topics are observed to have high probabilities for this hashtag. For all the three example hashtags, the topical keywords of the top-3 topics with the highest probabilities are listed in Table 4.5.

4.4 Applications

In this section, in order to illustrate the effectiveness of our models in addressing practical problems in Twitter, we apply our models to two problems as case studies. The two problems, namely *Hashtag Ranking for Tweet Annotation* and *Related Hashtag Discovery*, are first motivated and then the experimental results are presented.

4.4.1 Hashtag Ranking for Tweet Annotation

Hashtag facilitates tweet search and information diffusion. However, hashtags are annotated in only about 10% of tweets, observed from our data and also reported in other studies [HCC11]. Hashtag recommendation (same as hashtag ranking) attracts many researchers' attention. Due to that Twitter is widely considered as a real-time media, most methods for hashtag recommendation aim at online recommendation. More specifically, given a newly posted tweet, online recommendation is to recommend one or more hashtags immediately. However, the historical data accumulated in Twitter remains an important and rich information source for more advanced tweet

search options and other applications like retrospective event detection. Annotating historical tweets also helps to finding relevant tweets for less popular hashtags. As a case study, we evaluate the effectiveness of our models in *Hashtag Ranking for Tweet Annotation* which aims to annotate existing tweets without hashtags. More specifically, given a tweet d published by user u at time t , the task of hashtag ranking for tweet annotation is to annotate this tweet with the most appropriate hashtag(s). That is, we recommend hashtags to historical tweets. Next, we present the baseline method proposed in [KHLZ12] and discuss the solutions using our models.

Baseline methods: CF and CFU. In [KHLZ12], a Collaborative Filtering (CF) based method is proposed to recommend hashtags of a given tweet by considering both the tweet content and the user. Given a tweet d , by computing content similarity (*e.g.*, cosine similarity), the method finds the top- x most similar tweets with hashtags. The most frequent hashtags in these top- x tweets are then recommended. This method is named as CF in our study. Furthermore, in [KHLZ12], Kywe *et al.* presented a method considering additional user factor. We call Kywe’s method CFU method. In CFU , each user is formed as a hashtag vector. The hashtag vector is weighted by the $TF - IDF$ scheme, where the TF is the number of times this user has used a hashtag in all her tweets, and IDF is computed from the number of distinct users who have used this hashtag. With this hashtag vector, the authors retrieved the top- y most similar users to a user u . Then the hashtag to be recommended to a tweet d by user u is based on (i) the number of times a hashtag is used to annotate the top- x most similar tweets (from all users), and (ii) the number of times a hashtag has been adopted by the top- y most similar users.

Our proposed methods: CFU+CPM and CFU+HPM. The three factors user, time, and tweet content for hashtag annotation are modeled in both CPM and HPM. Given a tweet d written by user u at time t , the two models are able to directly estimate $p(h|u, t, \mathbf{w}_d)$. For tweet annotation problem, it is straightforward to rank hashtags by this probability. This method, however, delivers poorer accuracy than the

baseline methods. The reason is that many hashtags are under-represented because of their very limited usage in tweets. Recall that, the usage of hashtag follows a power-law like distribution (see Figure 4.3(a)) and most hashtags are used to annotate a small number of tweets, making the estimation $p(h|u, t, \mathbf{w}_d)$ less accurate for these hashtags.

We combine the recommendation by our models and the recommendation by the baseline methods to address the issue. Generally speaking, the combined method recommends hashtags by considering both the global factors (*i.e.*, the latent relationship between hashtag and user, time, and tweet content based on our models) and the local factors (*i.e.*, the most similar tweets and most similar users based on the baseline methods). In this following, we use $CFU + CPM$ as an example to illustrate the combined method.

Let r_h be the number of times a hashtag h is recommended by the baseline method CFU for tweet d . Let $p_n(h|u, t, \mathbf{w}_d)$ be the normalized recommendation score from CPM:

$$p_n(h|u, t, \mathbf{w}_d) = \frac{p(u, t, \mathbf{w}_d, h)}{\sum_{h'} p(u, t, \mathbf{w}_d, h')}$$

where the joint probability $p(u, t, \mathbf{w}_d, h) = \sum_z p(u, t, z, \mathbf{w}_d, h)$ and $p(u, t, z, \mathbf{w}_d, h)$ can be estimated with Equation 4.2 by replacing $\mathbf{h}_d$ with h in the equation. For HPM, $p(u, t, \mathbf{w}_d, h)$ is computed in a similar manner based on the joint probability $p(u, t, \mathbf{h}_d, z, \mathbf{w}_d)$ defined in Equation 4.11.

The recommendation score of hashtag h , denoted by $Score(h)$, by the combined method $CFU + CPM$ is:

$$Score(h) = \log(r_h + 1) \times p_n(h|u, t, \mathbf{w}_d) \quad (4.15)$$

In the above equation, the logarithm function is introduced to reduce the impact of extremely popular hashtags. Note that, if a hashtag h does not receive any recommendation from CFU , then $Score(h) = 0$ and this hashtag will not be recommended.

(a) Hit rate of top-5 hashtags

(b) Hit rate of Top-10 hashtags

Figure 4.7: Hit rate of the four methods for top-5/top-10 hashtags

Experimental Setting. We randomly select 200,000 tweets as test set and the hashtags adopted by these tweets are considered as the ground truth. We use *Hit Rate* to evaluate the annotation accuracy. Given a tweet, a hit occurs if at least one of the top- n recommended hashtags matches the ground truth hashtags of the tweet. The hit rate for a method is computed by the number of hits divided by the number of test tweets. We report the hit rate for top-5 and top-10 recommendations for all methods. We evaluated six methods in total: CF , CFU , $CFU + CPM$, $CFU + CPM^{sn}$, $CFU + HPM$, and $CFU + HPM^{sn}$.

Experimental Results. Recall that in CFU , top- x most similar tweets and top- y most similar users are retrieved for hashtag recommendation. In our experiments, we set x and y to be the same and evaluated 4 settings: $x = y = 5, 10, 15$, or 20 . The hit rates of top-5 and top-10 recommendations are reported in Figures 4.7(a) and 4.7(b) respectively for the six methods. We make the following three observations from the results.

First, for both top-5 and top-10 hashtag recommendations, the methods with either CPM or HPM perform better than both baseline methods CF and CFU . In particular, in terms of hit rate for top-5 hashtag recommendation with 5 similar tweets/users, $CFU + CPM$ outperforms CFU by 6.72% and CF by 14.34% respectively. We also observe that $CFU + HPM$ yields very similar results as $CFU + CPM$, despite that HPM achieves better perplexity than CPM in our earlier experiments.

Second, $CFU + CPM^{sn}$ performs slightly worse than $CFU + CPM$ and the same observation holds for $CFU + HPM^{sn}$ against $CFU + HPM$. In other words, considering social network regularization does not improve the hit rate for hashtag recommendation. One possible reason is that the social network regularization introduces noises in estimating $p(z|u)$. Consequently, the poorer estimation of $p(z|u)$ results in less accurate $p(h|u, t, \mathbf{w}_a)$. This result is consistent with the results reported in Section 4.3.2 where the considering social network regularization leads to poorer perplexity to both models.

Third, evaluated by hit rate of top-5 hashtag recommendation, the hit rate for all methods decreases along with increasing the number of similar tweets/users. This observation suggests a larger number of similar tweets/users likely brings in irrelevant hashtags to the given tweet, particularly when the ground truth hashtag is an infrequent hashtag. Recall that hashtag frequency distribution follows a power-law like distribution and a large number of hashtags appear only 5 times in our dataset (see Section 4.3.1). Take the tweet “How the PAP is going to help poor Singaporeans #sgpresident” as an example, when we search the content “How the PAP is going to help poor Singaporeans” with 5 similar tweets and users, the returned hashtag with the highest frequency in similar tweets and users is #sgpresident, and thus the recommendation is successful. However, when the number of similar tweets and users increases to 20, the returned hashtag with the highest frequency is #singapore and #sgpresident has a much lower rank. Therefore, the introducing irrelevant hashtags hurt the hit rate.

4.4.2 Related Hashtags Discovery

Hashtags are chosen by Twitter users from an uncontrolled vocabulary. Multiple hashtags might be chosen by users to target on the same event or the same topic, *e.g.*, #cikm, #cikm14, or #cikm2014 can be used for representing the same conference. Because of other types of relationships (*e.g.*, subsumption relation), hashtags are possible to be relevant to each other. For instance, the hashtag #sgelections has been used to annotate tweets relevant to both the Singapore Parliamentary General Election⁵ in May 2011 and the Singaporean Presidential Election⁶ in August 2011, while Twitter users also adopted a more specific hashtag #sgpresident for the latter. Discovering related hashtags helps users in refining, extending or reformulating hashtag-based queries.

⁵http://en.wikipedia.org/wiki/Singaporean_general_election,_2011

⁶http://en.wikipedia.org/wiki/Singaporean_presidential_election,_2011

Specifically, given a hashtag h and hashtag vocabulary E , related hashtag discovery is to locate the top- n hashtags from E (without h itself) that are most related to h . In this set of experiments, we evaluate four methods for their effectiveness in finding most related hashtags of a given hashtag.

Co-occurrence (*COO*). Using co-occurrence is a straightforward method for finding the related hashtags. If a hashtag h' often co-occurs with the given hashtag h in tweet annotation, then h' is believed to be related with h .

Content-based Similarity (*CBS*). If two hashtags share similar semantic meanings defined by the sets of tweets annotated by them, we consider these two hashtags are related. Given a hashtag h , a virtual document is formed by all tweets annotated by h . Then we compute the similarity of two hashtags based on the cosine similarity of the two corresponding virtual documents.

Content- and Topic-based Similarity (*CTS*). In this method, we use the topic-based feature representation to enhance the hashtag similarity computation. More specifically, each hashtag can be represented by a topic vector, where each dimension is one of the K topics and is weighted by $p(z_i|h)$, $0 \leq i \leq K$. Let $S_c(h, h')$ be the content-based similarity between hashtags h and h' computed in CBS, and let $S_t(h, h')$ be the cosine similarity between the topic vector representations of the two hashtags. The CTS similarity between the two hashtags is: $S_{ct}(h, h') = \eta \times S_c(h, h') + (1 - \eta) \times S_t(h, h')$, where η is a parameter for the combination. The following question is: how to compute $p(z|h)$ using the two models?

- In CPM, $p(z|h) = \frac{\sum_{d \in D_h} p(z|d)}{|D_h|}$ where $p(z|d)$ is computed using Equation 4.6.
- In HPM, $p(z|h)$ is estimated directly from the model (see Equation 4.12).

To summarize, we have four methods for evaluation: *COO*, *CBS*, CTS_{CPM} , and CTS_{HPM} where for the latter two CPM and HPM denote the model for computing the topic vector for hashtags.

Table 4.6: Kappa scores between three pairs of volunteers (v 's)

Volunteer pair	$\langle v_1, v_2 \rangle$	$\langle v_1, v_3 \rangle$	$\langle v_2, v_3 \rangle$	Average
Kappa score	0.809	0.687	0.672	0.723

Table 4.7: Precision for related hashtag discovery with the best result in boldface

Method	<i>COO</i>	<i>CBS</i>	<i>CTS_{CPM}</i>	<i>CTS_{HPM}</i>
Precision	0.520	0.681	0.705	0.729

To evaluate the effectiveness of the four methods in finding related hashtags, we randomly selected 50 hashtags among the top-500 most popular hashtags to be the query hashtags.⁷ For each of the 50 query hashtags, a method returns the top-5 most related hashtags for manual assessment. In CTS, η is set to 0.6 in our experiments based on observations using a few sample hashtags (not included in the 50 query hashtags). We employ three volunteers to label the relatedness of the top-5 hashtags returned by each method and each hashtag receives a binary score: 0 for not-related and 1 for related. The kappa scores of the agreement between any pair of the volunteers are reported in Table 4.6. The average kappa score is 0.723 suggesting substantial agreement between our volunteers.

The average precision for the 50 query hashtags from the three volunteers is reported in Table 4.7. Note that *COO* yields the poorest precision. One reason is that there are 82.2% of tweets annotated with only one hashtag (see Section 4.3.1). For a given query hashtag, there are too few co-occurring hashtags. Among the other three methods, which use content similarity, *CTS_{CPM}* and *CTS_{HPM}* outperform the method not using topic vector. This evidences the effectiveness of using topic vector as additional information for the problem of related hashtag discovery. Note that *CTS_{HPM}* achieves the highest precision, one possible reason is that HPM discovers more meaningful topics reflected by the lowest perplexity (see Section 4.3.2).

We now use two examples hashtags *#prayforjapan* and *#movies* to illustrate the difference between the most related hashtags found by the four methods, listed in Table 4.8. Among the top-5 most related hashtags for *#prayforjapan* found by *COO*,

⁷Popular hashtags are expected to have higher chances of being co-occurred with other hashtags.

Table 4.8: Top-5 most related hashtags to #prayforjapan and #movies, discovered by the four methods

Method	Top related hashtags to #prayforjapan
<i>COO</i>	#japan #prayfortheworld #tsunami #fb #sleague
<i>CBS</i>	#japan #tsunami #quake #fukushima #japans
<i>CTSCPM</i>	#prayfortheworld #japan #helpjapan #godblessjapan #quake
<i>CTSHPM</i>	#prayfortheworld #helpjapan #japan #godblessjapan #quake
-	Top related hashtags to #movies
<i>COO</i>	#imdb #singapore #sg #singaporean #film
<i>CBS</i>	#imdb #celebrity #gossip #xinmsn #ryanreynolds
<i>CTSCPM</i>	#imdb #movie #mfpgossip #seattle #eastboundanddown
<i>CTSHPM</i>	#imdb #movie #sgfilm #trailer #video

#fb and #sleague are not related. All the remaining three methods *CBS*, *CTSCPM*, and *CTSHPM* are able to find related hashtags for #prayforjapan. Interestingly, the two methods with topic-level representation recommend the same set of hashtags in slightly different orders. Another example is #movies. All top-5 hashtags by *CTSHPM* are relevant to #movie. The hashtags from the other three methods all contain some irrelevant hashtags such as #sg, #xinmsn and #eastboundanddown.

4.5 Summary

In this chapter, we propose two PLSA-style topic models to model the latent relationship between tweet content, user interest, time, and hashtag at topic-level. We also evaluate the impact of considering social network regularization based on mention relationship in Twitter. Through extensive experiments, we show that Hashtag-Pivoted Model outperforms Content-Pivoted Model in terms of perplexity measure. We also show that the social network regularization based on mention relationship hurts the performance of both models. We further demonstrate the effectiveness of the two models in addressing two practical applications (*i.e.*, hashtag ranking for tweet annotation and related hashtag discovery). The utilization of both models improves the effectiveness in addressing both applications compared to their corresponding baselines.

Chapter 5

Answer Ranking for Questions

In this chapter, we study domain-specific CQA systems (*e.g.*, Stack Overflow, Med-help), which are different from CQA systems for general topics (*e.g.*, Yahoo! Answers, Baidu Knows). Questions and answers in domain-specific CQA systems are mostly in the same topical domain, enabling more comprehensive interaction between users on fine-grained topics. A question often has multiple answers. We call question *master document* and answer *slave document*. In CQA systems, users are more likely to ask questions on unfamiliar topics and to answer questions matching their expertise. Users can also vote answers based on their judgements. In this chapter, we propose a Tri-Role Topic Model (TRTM) to model the tri-roles of users (*i.e.*, as askers, answerers, and voters, respectively) and the activities of each role including composing question, selecting question to answer, contributing and voting answers. The proposed model can be used to enhance CQA systems from many perspectives. As a case study, we conducted experiments on ranking answers for questions on Stack Overflow, a CQA system for professional and enthusiast programmers. Experimental results show that TRTM is effective in facilitating users getting ideal rankings of answers, particularly for new and less popular questions. Evaluated on nDCG, TRTM outperforms state-of-the-art methods.

The rest of this chapter is organized as follows. Section 5.1 briefly review the related work. Section 5.2 presents the Tri-Role Topic Model and its inference algo-

rithm. Section 5.3 evaluates the proposed models measured by discovered topics and applies TRTM to the problem of answer ranking for questions. Section 5.4 summarizes this chapter.

5.1 Related Work

Stack Overflow. Anderson *et al.* ([AHKL12]) found that in Stack Overflow, expert users are likely to answer questions more quickly, and a higher activity level of a question benefits all answerers of this question to increase their reputation level. Based on some extracted features from Stack Overflow, they attempted to predict the long-term value of a question and whether a question has been sufficiently answered. Their results show that votes indicate a user’s expertise level on a specific topic. This is consistent with the modeling of votes in our proposed TRTM. Subsequently, Anderson *et al.* ([AHKL13]) observed that badge mechanism in Stack Overflow steers users to increase their participation in answering questions. Question deletion in Stack Overflow was studied in [CS14], where 47 features were used to predict whether a question will be deleted. The quality of question content is found to be the main factor.

Dalip *et al.* ([DGCC13]) proposed to rank answers of a question in Stack Overflow using Learning to Rank (L2R), a supervised approach. The L2R model is learned from the feature vector representations of question-answer pairs. Each pair is represented by features in 8 groups (*e.g.*, user features, structure features, style features). Answers of new coming questions are then predicted by the trained L2R model. Note that, L2R is a supervised approach. In our work, we focus on a generative probabilistic model, which is unsupervised in nature, to model the three roles of users and their activities.

Table 5.1: Notations for TRTM

Symbol	Description
U	Collection of askers $u \in U$
V	Collection of answerers $v \in V$
Q	Collection of questions $q \in Q$
A	Collection of answers $a \in A$
Z	Collection of topics $z \in Z$
W	Word vocabulary of questions
E	Word vocabulary of answers
Q_u	Set of questions composed by u
Q_v	Set of questions answered by v
A_v	Set of answers contributed by v
V_q	Set of answerers to question q
s_q^a	Voting score of answer a to question q
$n(q, w)$	Number of word $w \in W$ in question q
$n(a, e)$	Number of word $e \in E$ in answer a

5.2 Tri-Role Topic Model

We start with the notations used in our model, summarized in Table 5.1. We then present the Tri-Role Topic Model (TRTM) and its inference algorithm.

5.2.1 Notations

Let Q and A be the set of questions and the set of answers respectively. W is the word vocabulary of questions and E is the word vocabulary of answers. Let U be the set of askers (*i.e.*, users who ever asked questions), and let V denote the set of answerers (*i.e.*, users who ever answered questions). Note that, a user can be an asker for one question and be an answerer of another question, *i.e.*, $U \cap V \neq \Phi$. We further use Q_u to denote the set of questions composed by asker u , and use Q_v to denote the set of questions answered by answerer v . The set of answers contributed by v is denoted by A_v .

An answer a , contributed by answerer v to question q , may receive zero or more votes. As a vote can be either positive or negative, the aggregated number of votes to an answer may be negative. For easy processing, we compute a *voting score* for

Figure 5.1: The tri-roles of a user and the activities

answer a to question q as $s_q^a = x - x_{min} + 0.5$, where x is raw aggregated vote for this answer and x_{min} is the lowest aggregated vote of an answer in our data collection, and 0.5 is a constant to ensure $s_q^a > 0$. In other words, s_q^a is the aggregated vote of an answer shifted to positive region.

5.2.2 Model Description

In TRTM, each user has three roles: an asker, an answerer, and a voter (see Figure 5.1). As discussed earlier, different from a question or an answer, a vote is not associated with any textual content. Moreover, votes become meaningful only when the number of votes is large. We therefore do not model the voter role explicitly in TRTM. Instead, voting is used to constrain the topic distributions in our model.

TRTM models four types of topic distributions. Let z denote topic. The four types of topic distributions are: (i) $p(z|u)$, topic distribution of asker u , (ii) $p(z|v)$, topic distribution of answerer v , (iii) $p(z|q)$, topic distribution of question q , and (iv) $p(z|a)$, topic distribution of answer a . Note that, TRTM assumes that each question or answer has multiple topics and each word is sampled from its corresponding topic with probability $p(w|z)$ or $p(e|z)$. Regarding topic distributions, we make the following three assumptions (**A1**, **A2**, and **A3**) in TRTM:

- **A1:** An asker u and all the questions composed by her Q_u share similar topic distributions.
- **A2:** An answerer v and all the questions answered by her Q_v , share similar topic distributions. The degree of similarity in the topic distributions between v and Q_v is reflected by the voting scores of her answers. If an answer to question q by v receives a large voting score, then most users believe that this answer well addresses the question; hence answerer v has the expertise in answering this question q . v and q therefore share more similar topic distributions. Here, we use the *voter* role of users (*i.e.*, those who are not answerers or the asker) to constrain the topic distributions between the answerer and her questions. In simple words, an answerer's topic distribution is more similar to that of the questions answered by her, if her answers to these questions receive more positive votes.
- **A3:** An answerer v and all the answers contributed by her A_v share similar topic distributions.

In TRTM, we adopt the *exponential KL-divergence* (eKL) function to model the relationship between two topic distributions. Proposed in [KPS13], the eKL function is the combination of exponential probability densities¹ and KullbackLeibler divergence². For two k -dimensional probability distributions μ and θ and a given scalar λ , $eKL(\theta, \lambda, \mu)$ is defined as:

$$eKL(\theta, \lambda, \mu) = \lambda e^{-\lambda KL(\mu||\theta)}$$

where $KL(\mu||\theta)$ is $\sum_k \mu_k \log(\mu_k/\theta_k)$. The properties of the $eKL(\theta, \lambda, \mu)$ function include: (i) with a fixed λ , the eKL value increases with the degree of similarity between μ and θ , and (ii) with a larger λ , the exponential probability densities decrease faster when increasing the value of $KL(\mu||\theta)$.

¹http://en.wikipedia.org/wiki/Exponential_distribution

²http://en.wikipedia.org/wiki/Kullback-Leibler_divergence

Figure 5.2: Tri-Role Topic Model

The generative process of TRTM is divided into three sub-procedures (see Figure 5.1), namely (i) composing question, (ii) selecting question to answer, and (iii) contributing answer.

Composing question. An asker u composes a question $q \in Q_u$ with probability $eKL(q, \alpha, u)$, where α is a scalar. When topic distributions of q and u are more similar, u is more likely to compose question q because of interest matching (Assumption **A1**). Next, for each word w in q , a topic z is sampled with probability $p(z|q)$, and then w is generated based on $p(w|z)$.

Selecting question to answer. The probability of an answerer v choosing to answer a question $q \in Q_v$ is modeled as $eKL(q, \beta \cdot s_q^a, v)$, where s_q^a is the voting score and β is a scalar. Recall that in Assumption **A2**, the degree of similarity in the topic distributions between v and Q_v is reflected by the voting scores of her answers. The larger the voting score s_q^a , the sharper the curve of $eKL(q, \beta \cdot s_q^a, v)$, which means that the eKL assigns a higher probability when the distance between v and q gets smaller.

Contributing answer. An answerer v contributes an answer $a \in A_v$ with probability $eKL(a, \tau, v)$, where τ is a scalar. Based on Assumption **A3**, v prefers to contribute a

if the topical similarity between v and a is high. Next, for each word e in a , a topic z is sampled with probability $p(z|a)$, and then e is generated with $p(e|z)$.

The graphical representation of the TRTM model is shown in Figure 5.2 and the generative process is summarized as follows:

- For each question-answer pair (q, a) , where $q \in Q_u$ and $a \in A_v$
 - Asker u composes question q with probability $eKL(p(z|q), \alpha, p(z|u))$
 - For each word w in q
 - * Draw a topic z from $p(z|q)$
 - * Draw a word w from $p(w|z)$
 - Answerer v selects to answer question q with probability $eKL(p(z|q), \beta \cdot s_q^a, p(z|v))$
 - Answerer v contributes answer a with probability $eKL(p(z|a), \tau, p(z|v))$
 - For each word e in a
 - * Draw a topic z from $p(z|a)$
 - * Draw a word e from $p(e|z)$

TRTM is extended from the PLSA model. In PLSA, only the topic distribution of posts is modeled, while TRTM is able to model the topic distributions of posts and users simultaneously with the probability $eKL(p(z|q), \alpha, p(z|u))$ and $eKL(p(z|a), \tau, p(z|v))$. Without the eKL probability, the topic distributions of the asker and the answerer will be ignored and there exist two PLSA models in TRTM for the generation of questions and answers. Furthermore, we utilize the probability $eKL(p(z|q), \beta \cdot s_q^a, p(z|v))$ to connect the generation of topic distribution of questions with that of answers, which indicates that these two topic distributions are regularized by the voting scores.

5.2.3 The Inference Algorithm of TRTM

Given question set Q composed by U and answer set A by answerers from V , we obtain the likelihood of the data in Equation 5.1.

$$\begin{aligned}
 L = & \prod_{u \in U} \prod_{q \in Q_u} eKL\left(p(z|q), \alpha, p(z|u)\right) \\
 & \prod_{u \in U} \prod_{q \in Q_u} \prod_{w \in W} \left[\sum_z p(w|z)p(z|q) \right]^{n(q,w)} \\
 & \prod_{v \in V} \prod_{q \in Q_v} eKL\left(p(z|q), \beta \cdot s_q^a, p(z|v)\right) \\
 & \prod_{v \in V} \prod_{a \in A_v} eKL\left(p(z|a), \tau, p(z|v)\right) \\
 & \prod_{v \in V} \prod_{a \in A_v} \prod_{e \in E} \left[\sum_z p(e|z)p(z|a) \right]^{n(a,e)}
 \end{aligned} \tag{5.1}$$

The exact inference of Equation 5.1 is intractable. We propose an Expectation-Maximization (EM) algorithm to appropriately infer TRTM. The EM algorithm has two steps: E-step and M-step. The E-step calculates the expectation of the hidden variables *i.e.*, $p(z|q, w)$ and $p(z|a, e)$ in TRTM.

E-step:

$$\begin{aligned}
 p^{k+1}(z|q, w) &= \frac{p^k(w|z)p^k(z|q)}{\sum_{z' \in Z} p^k(w|z')p^k(z'|q)} \\
 p^{k+1}(z|a, e) &= \frac{p^k(e|z)p^k(z|a)}{\sum_{z' \in Z} p^k(e|z')p^k(z'|a)}
 \end{aligned}$$

The M-step maximizes the log-likelihood (see Equation 5.1). The following probabilities are calculated: $p(w|z)$, $p(e|z)$, $p(z|u)$, $p(z|q)$, $p(z|a)$, and $p(z|v)$.

M-step:

$$\begin{aligned}
 p^{k+1}(w|z) &= \frac{\sum_{q \in Q} n(q, w) p^{k+1}(z|q, w)}{\sum_{w' \in W} \sum_{q \in Q} n(q, w') p^{k+1}(z|q, w')} \\
 p^{k+1}(e|z) &= \frac{\sum_{a \in A} n(a, e) p^{k+1}(z|a, e)}{\sum_{e' \in E} \sum_{a \in A} n(a, e') p^{k+1}(z|a, e')} \\
 p^{k+1}(z|u) &= \frac{[\prod_{q \in Q_u} p^{k+1}(z|q)]^{1/|Q_u|}}{\sum_{z' \in Z} [\prod_{q \in Q_u} p^{k+1}(z'|q)]^{1/|Q_u|}}
 \end{aligned}$$

$$\begin{aligned}
 p^{k+1}(z|v) &= \frac{[\prod_{q \in Q_v} p^{k+1}(z|q)^{\beta \cdot s_q^a} \prod_{a \in A_v} p^{k+1}(z|a)^\tau]^{1/(\sum_{q \in Q_v} \beta \cdot s_q^a + \tau |A_v|)}}{\sum_{z' \in Z} [\prod_{q \in Q_v} p^{k+1}(z'|q)^{\beta \cdot s_q^a} \prod_{a \in A_v} p^{k+1}(z'|a)^\tau]^{1/(\sum_{q \in Q_v} \beta \cdot s_q^a + \tau |A_v|)}} \\
 p^{k+1}(z|q) &= \frac{\sum_{w \in W} n(q, w) p^{k+1}(z|q, w) + \sum_{u \in U_q} \alpha p^k(z|u) + \sum_{v \in V_q} \beta \cdot s_q^a \cdot p^k(z|v)}{\sum_{z' \in Z} \{ \sum_{w \in W} n(q, w) p^{k+1}(z'|q, w) + \sum_{u \in U_q} \alpha p^k(z'|u) + \sum_{v \in V_q} \beta \cdot s_q^a \cdot p^k(z'|v) \}} \\
 p^{k+1}(z|a) &= \frac{\sum_{e \in E} n(a, e) p^{k+1}(z|a, e) + \sum_{v \in V_a} \tau p^k(z|v)}{\sum_{z' \in Z} \{ \sum_{e \in E} n(a, e) p^{k+1}(z'|a, e) + \sum_{v \in V_a} \tau p^k(z'|v) \}}
 \end{aligned}$$

We iteratively compute probabilities of **E-step** and **M-step** until achieving convergent log-likelihood (see Equation 5.1). k represents the k th iteration of *EM* algorithm. Note that, in **M-step**, we first calculate $p(z|q)$ and $p(z|a)$, then calculate $p(z|u)$ and $p(z|v)$.

5.2.4 Complexity Analysis

We now analyze the time complexity of our proposed TRTM model. In the E-step, the time complexity is $O(TK(|Q||W_q^{avg}| + |A||E_a^{avg}|))$, where T , K , $|Q|$, $|A|$, $|W_q^{avg}|$, $|E_a^{avg}|$ indicate the number of iterations, the number of topics, the number of questions, the number of answers, the average number of words in question q and the average number of words in answer a . In the M-step, the time complexity is $O(TK(|W| + |E| + |U| + |V| + |Q| + |A|))$, where $|W|$, $|E|$, $|U|$, $|V|$ are the vocabulary size of questions, the vocabulary size of answers, the number of askers and the number of answerers, respectively. The space complexity for TRTM is $O(K(|Q||W_q^{avg}| + |A||E_a^{avg}| + |W| +$

$|E|+|U|+|V|+|Q|+|A|$). Observe that all the space is used to store the probabilities in the E-step and M-step.

5.3 Experiments

We evaluate the proposed TRTM model on Stack Overflow data³. We report one case study for the application of answer ranking for questions.

Data Set. Questions and answers from Stack Overflow posted between 01 January 2011 and 31 March 2011 are used as training data; questions and answers published from 01 April 2011 to 06 September 2013 are used as test data.

We preprocess the training data by removing questions and answers from inactive users. More specifically, a user is inactive if the total number of questions and answers posted by her is smaller than 80, as defined in [YQG⁺13]. After preprocessing, the training data contains 16,141 questions from 868 askers and 180,394 answers from 1,184 answerers. Note that a user could play a single role (*i.e.*, an asker or an answerer). The vocabulary size for questions is 21,760 and that for answers is 85,889. The raw aggregated vote is in the range of -10 to 359.

For test data, questions with fewer than 5 answers are removed, because answer ranking is more meaningful if a question has a large number of answers. As the result, the test data contains 20,834 questions and 150,320 answers. Note that, the askers and answerers in the test data may not appear in the training data.

We experimentally set the hyperparameters of TRTM: $\alpha = 100$, $\beta = 100$, $\tau = 100$. We evaluated different number of topics $|Z| = 10, 20$, and 40.

5.3.1 Topic Discovery

Discovered by TRTM, we randomly select 5 topics of questions and 5 topics of answers as examples, shown in Tables 5.2(a) and 5.2(b) respectively. The top-10 words

³<http://blog.stackoverflow.com/category/cc-wiki-dump/>

Table 5.2: Example topics by TRTM from Stack Overflow

(a) Example question topics with topic ID and words

ID	Top-10 words with highest generative probability
6	file error window install reference build directory js header include
8	app java process compile org template apache module map default
12	image project script photo folder null get collection assembly generate
14	php div tag run html load link content element use
18	data create database select custom ve use answer please size

(b) Example answer topics with topic ID and words

ID	Top-10 words with highest generative probability
1	change memory compile program language address python stack git branch
5	string page example html look jquery instead url javascript content
8	application java request access cache compile load log map api
13	thread bit process result field template loop message format single
20	name server service net project client source standard connection header

based on $p(w|z)$ and $p(e|z)$ respectively are listed for each topic. Observe that TRTM captures some major topics of questions and answers in Stack Overflow. Questions related to **Java programming** (Topic 8) and **Web development** (Topic 14) are frequently asked. **Process and thread** (Topic 13) and **Server client programming** (Topic 20) are prevalent in answers. Note that, the vocabulary sets of the questions and of the answers are significantly different. Naturally, the number of words in all answers to a question is much larger than the words in the question itself. More importantly, there are more technical terminologies in answers than that in questions.

We also show topic distributions of a randomly selected user for her two roles: as an asker and as an answerer in Figure 5.3 and Table 5.3. Observe that the topic distributions for this user as the two roles are significantly different. For instance, topics 11 and 16 have large $p(z|u)$ probability values in her question topic distribution, which indicates that the example user is eager to ask questions about array implementation and event handler. She might ask questions like “how to remove all event handlers from a control?”, and our model regularizes the topic distribution of the asker role and the topic distribution of her questions to converge by the eKL function. For the answer topic distribution, topic 15 has the largest probability $p(z|v)$, which in-

Figure 5.3: Topic distributions of asker role and answerer role from the same example user

Table 5.3: Topics of the same example user

ID	Top-10 words with highest generative probability
Question Topic 11	question example array implement look arraylist api foo doesn bar
Question Topic 16	time change event text click button field date result version
Answer Topic 15	re don store ll block hash probably output input actually

indicates that the example user is an expert at the topic on data storage. She might have answers like “My suggestion is to use MS Access database as your JAR’s data keeper.”, and our model regularizes the topic distribution of the answerer role and the topic distribution of her answers to converge by the eKL function. Furthermore, we could infer that the example user prefers to answering questions about data storage. Through the analysis of the question topic distribution and answer topic distribution of a specific user, we could easily evaluate what topic the user is good at and what topic the user is eager to learn. TRTM is capable of distinguishing question and answer topic distributions for users as askers and answerers.

5.3.2 Answer Ranking for Questions

A popular question could receive many answers within a very short period. However, given the short time period, there might be lack of enough votes to help the asker to select the high quality answers, because the answers may be from a domain the asker

is unfamiliar with. Timely ranking answers for questions benefits askers in quickly getting high-quality answers.

Problem definition. Given a question q and its answer set A_q , the task of *ranking answers* is to rank answers $a \in A_q$ such that the top-ranked answers best address q . In this sense, we assume that the best answers for a question are the ones sharing most similar topic distributions with the question. The answers are then ranked by topical similarities to the question, and the topics of a 's and q are learned using topic models, TRTM or other baseline models.

The *topical similarity* (TS) between a question q and an answer a is evaluated using Jensen-Shannon divergence,

$$TS(q, a) = JSD(\theta_q, \theta_a)$$

where θ_q and θ_a represent the topic distributions of question q and answer a respectively.

$$\begin{aligned}\theta_q &\approx p(\mathbf{w}_q|z) = \sum_{w \in \mathbf{w}_q} p(w|z) \\ \theta_a &\approx p(\mathbf{e}_a|z) = \sum_{e \in \mathbf{e}_q} p(e|z)\end{aligned}$$

In above equations, $\mathbf{w}_q$ and $\mathbf{e}_q$ are word vectors for question q and answer a respectively.

Baseline methods. Latent Dirichlet Allocation (**LDA**) is a standard technique for topic analysis in document collections [BNJ03]. Here, a virtual document is created for each user by aggregating all her questions and answers, and then LDA is employed to learn the hidden topics (*i.e.*, $p(w|z)$). Topic Expertise Model (**TEM**) is a very recent model proposed in [YQG⁺13]. Considered as a state-of-the-art baseline, TEM jointly models user topical interests and expertise in a probabilistic model. TEM has

Table 5.4: nDCG of the three models; the best result for each topic number setting ($|Z|=10, 20, 40$) is in boldface.

$ Z $	Model	$nDCG@1$	$nDCG@5$	$nDCG@10$	$nDCG$
10	TRTM	0.3273	0.6448	0.6759	0.6762
	TEM	0.3005	0.6281	0.6607	0.6611
	LDA	0.3026	0.6296	0.6618	0.6622
20	TRTM	0.3405	0.6518	0.6824	0.6828
	TEM	0.3052	0.6303	0.6630	0.6633
	LDA	0.3093	0.6331	0.6651	0.6654
40	TRTM	0.3380	0.6506	0.6806	0.6810
	TEM	0.3106	0.6333	0.6657	0.6660
	LDA	0.3195	0.6411	0.6719	0.6722

been evaluated on Stack Overflow data and has been applied for the task of answer ranking but with a different problem setting in [YQG⁺13]⁴. For both LDA and TEM, the topical similarity is computed in a similar way as in TRTM.

Evaluation measure. We use normalized discounted cumulative gain (nDCG) to evaluate the list of ranked answers, following [YQG⁺13]. Here, the ground-truth ranking of the answers to a question is the ranking by the number of aggregated votes of the answers. The number of aggregated votes is also used to define the degree of relevance of each item in a rank, required by the nDCG measure. $nDCG@M$ for the top- M ranked answers of test question q computed as follows:

$$nDCG(q, M) = \frac{1}{IDCG(q, M)} \sum_{i=1}^M \frac{2^{rv_{q,i}} - 1}{\log_2(i + 1)}$$

where $rv_{q,i}$ is the number of aggregated votes received by the answer ranked at the i -th position; $IDCG(M, q)$ is the normalization factor for the discounted cumulative gain of the ideal ranking of the top- M answers for question q . Then $nDCG@M$ is the average of $nDCG(q, M)$ over all questions in the test data.

⁴In TEM, only the answers from the answerers that appear in training data are ranked. In our proposed solution, we utilize the words in an answer (*i.e.*, $p(z|w)$) where the answerer may not appear in the training data.

Experimental results. The $nDCG@M$'s of the three models with $|Z| = 10, 20$, and 40 topics are reported in Table 5.4 where $M=1, 5, 10$, and all answers. The following three observations are made from the results.

- TRTM performs better than both baseline methods TEM and LDA, for all different settings on number of topics and on all M settings. Particularly, on $nNCG@1$, TRTM outperforms TEM by 11.6% and LDA by 10.1% respectively. The results evidence the effectiveness of our proposed model.
- All models with 20 topics yield best results, which suggests that 20 is a more appropriate number of topics on this dataset. On the other hand, all the three models are relatively not very sensitive to topic number setting.
- LDA slightly outperforms TEM. One possible reason is that TEM assumes each question (resp. each answer) has only one unique topic, which is not appropriate in modeling Stack Overflow data, where some questions and answers are fairly long and may cover multiple topics.

5.4 Summary

In this chapter, we propose a Tri-Role Topic Model to model the tri-roles of users (*i.e.*, askers, answerers, and voters) in CQA systems and the activities of each role (*i.e.*, composing question, selecting question to answer, contributing, and voting answers). Our model is capable of mining four topic distributions of asker, answerer, question, and answer, respectively. We demonstrate the effectiveness of our model in discovering topics from Stack Overflow and also in addressing the problem of answer (slave document) ranking for questions (master documents) on the same dataset. Evaluated on $nDCG$, TRTM outperforms state-of-the-art methods.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The extreme popularity of UGC data evidences its effectiveness in sharing and broadcasting information among Web users. Because UGC is contributed by common Web users, in most cases, the data is written in informal style and is unstructured. As mentioned in Chapter 1, different from text data produced by traditional media, UGC shares three characteristics: (1) Huge Amount, (2) Free Writing Style, and (3) Heterogeneous Data. In addition, UGC is not organized in flat structure. The user-generated documents often demonstrate master-slave relationships and one master document may be associated with a large number of slave documents. Thus, UGC introduces not only convenience of information sharing for users, but also new issues like reading overload. Effectively and efficiently mining of UGC is a big challenge for many researchers. In this dissertation, we focus on the problem of ranking UGC data using topic models considering various factors in UGC data and the relationships among documents. More specifically, we model three types of UGC with master-slave document structure and aim to rank slave documents (comments, hashtags, answers) for their corresponding master document (news article, tweet, question). In order to obtain a better slave document ranking, we measure the similarity between slave documents and their corresponding master documents from topic level using topic

models extended from LDA and PLSA. Our experimental results demonstrate the effectiveness for the extended topic models in the three main tasks.

In Chapter 3, we define the problem of comment ranking for news articles. We aim to rank comments covering more topics and perspectives for a news article, which is the master document. The objective of the research is to present to users a summary of comments covering the major topics discussed among the readers, with respect to the topics covered in the master news article. After analyzing the important relationships between news and comments (*i.e.*, news-news, news-comment, comment-comment), we propose two generative models, namely, Master-Slave Topic Model and Extended Master-Slave Topic Model respectively, to implicitly model these relationships. Both models assume a slave document has exactly one topic. While MSTM constrains that the topic of a slave document has to be derived from its master document, EXTM allows the topic of a slave document derived from all slave documents. Evaluated using perplexity, in our experiments, EXTM significantly outperforms MSTM for grouping comments with similar topics. To generate the summary, we utilize two comment ranking schemes MMR and RL to select the most representative comments from each topical comment cluster. Our user study shows that EXTM-MMR outperforms the other five methods by topic-cohesion and topic-diversity measures.

In Chapter 4, we aim to rank slave documents best summarizing their corresponding master document and define the problem of hashtag ranking for tweet annotation. In order to address this problem, we propose two PLSA-style topic models (Content-Pivoted Model and Hashtag-Pivoted Model) to model the latent relationship between tweet content, user interest, time, and hashtag at topic-level. We also evaluate the impact of considering social network regularization based on mention relationship in the Twitter social network. Through extensive experiments, we show that HPM outperforms CPM in terms of perplexity measure. We also show that the social network regularization based on mention relationship adversely affects the performance of both models. We further demonstrate the effectiveness of the two models in addressing the problem of hashtag annotation in Twitter. The utilization of both models improves

the effectiveness in addressing the hashtag annotation problem compared to baseline methods.

In Chapter 5, we aim to rank answers as slave documents to their corresponding questions as master documents. The problem is defined as answer ranking for questions in domain-specific CQA systems. A Tri-Role Topic Model is proposed to model the tri-roles of users (*i.e.*, askers, answerers, and voters) in CQA systems, and the activities of each role (*i.e.*, composing question, selecting question to answer, contributing, and voting answers). Our model is capable of mining four topic distributions of asker, answerer, question, and answer, respectively. We demonstrate the effectiveness of our model in discovering fine-grained topics from Stack Overflow and also in addressing the problem of ranking answers for questions on the same dataset.

In summary, our research is mainly on three ranking problems of UGC data with the master-slave structure from different platforms. From Yahoo! News to Twitter, then to Stack Overflow, with the more complicated features of data adopted in our research, the proposed topic models include more features and relationships to simulate the generative process of the data. For the problem of comment ranking, the slave documents (comments) are much shorter than their corresponding master document (news article). Our main concern is discovering topics from comments which reflect the topics of their news article as well as keeping topics merely discussed among comments. For the problem of hashtag ranking, due to the extremely shortness of slave documents (hashtags), we introduce user and time factors to enrich the representation of hashtags for ranking hashtags more accurately. For the problem of answer ranking, the answers could receive votes from voters. It is a big challenge for us to model the voting behavior in a generative model. Besides introducing the user factor as in CPM and HPM, our TRTM model also introduces the voting factor to model the relationships between questions, answers, askers and answerers by the the exponential KL-divergence function.

6.2 Future Work

Ranking UGC remains a challenging issue. UGC plays a key role in posting and sharing information among users. Though UGC from different websites or platforms has different characteristics and structures, they might all face the ranking problem which causes users' reading overload. However, because of the different characteristics and structures of the UGC produced by different platforms, to model the topic distributions of such data, topic models are to be extended to consider platform specific features. With respect to the three platforms covered in this research, we recommend the following research directions.

For news websites allowing user comments, both our proposed models (*i.e.*, MSTM and EXTM) ignore the reply relationship. The reply relationship among comments provides rich structure information about comments. Because of the reply relationship, there exists a network among comments of a given news article. Comments replying to each other are more likely to share similar topics. An indicator generated from Beta distribution is able to model the existence of a link between two comments. In addition, the author of comments is an important factor to mine topics from comments, for the reason that comments of the same author are more likely to share similar topics. We recommend to incorporate reply relationship and the author factor in the two models to take advantage of the rich structural information in comments. We also observe that the topics discovered might be affected by some extremely long comments. Long comments possibly have multiple topics and perspectives. To improve our model for better handling long comments is also part of the future work.

For the platform of Twitter or other similar micro-blogging services, given a tweet, there is currently no mechanism to predict which model best reflects the generation process between its hashtag and tweet content. Research on such predicting mechanism is part of the future work. Another piece of future work is to evaluate the impact of social network regularization to the models based on other types of user relation-

ships than mention relationship, which might potentially improve the performance of ranking hashtag for tweets. For example, the following relationship is potentially a good factor to model topical relations between users. Compared with mention relationship, the graph of following relationship is relatively more dense, which may contain more information in social aspect. Detecting which tweet requires hashtags is also an interesting direction. A user who posts a tweet might not intend to use any hashtag at all. In that case, recommending hashtags for users is annoying. Therefore, filtering out tweets which do not require hashtags is a first step for hashtag ranking for tweets. A possible way for this hashtag-requirement detection task is to mine users' historical behavior of hashtag usage and tweet content.

For the platform of CQA systems, we expect to further model the rich temporal patterns of users in asking and answering questions. We observe that different users prefer to answer questions at different time point of a day and different day of a week. Incorporating temporal patterns of users has great potential in modeling users' activities more accurately. Another piece of future work is to introduce the badge mechanism into our model. The badge mechanism is used to evaluate users' expert level in Stack Overflow and some other domain-specific CQA systems. Thus, this prior knowledge is useful to model users' level of expertise. Users with different badges form multiple communities and supervised topic models is a direction for further exploration.

Furthermore, we expect to discover and analyze more common characteristics of UGC data from different platforms, which is helpful for us to design a more generalized topic model for mining multiple UGC data sets. For instance, a switch can be employed to model the relationship between users and posts for different types of UGC data. With the switch, we can more flexibly model the existence of the relationship between different features. In addition, the topic number in our models has to be set experimentally. In the future, we will employ Hierarchical Dirichlet Process [TJBB06] into our models to automatically learn the topic number.

Appendix A

List of Publications

Chapters 3 - 5 of this thesis are adapted based on the following publications during my PhD study.

- **Zongyang Ma**, Aixin Sun, Quan Yuan, Gao Cong. A Tri-Role Topic Model for Domain-Specific Question Answering. Proceeding of the 29th Conference on Artificial Intelligence (**AAAI 2015**). Austin, TX, USA. January 25 - 30, 2015.
- **Zongyang Ma**, Aixin Sun, Quan Yuan, Gao Cong. Tagging Your Tweets: A Probabilistic Modeling of Hashtag Annotation in Twitter. Proceedings of 23rd ACM Conference on Information and Knowledge Management (**CIKM 2014**), Pages 999-1008. Shanghai, China, November 3 - 7, 2014.
- **Zongyang Ma**, Aixin Sun, Quan Yuan, Gao Cong. Topic-Driven Reader Comments Summarization, Proceedings of the 21st ACM Conference on Information and Knowledge Management (**CIKM 2012**), 265-274, Maui, USA, October 29 - November 2, 2012.

I have also conducted research that are related to this thesis reported in the following publications.

- **Zongyang Ma**, Aixin Sun, Gao Cong. On Predicting Popularity of Newly Emerging Hashtags in Twitter. *Journal of the American Society for Information Science and Technology (JASIST)*, Volume 64, Pages 1399 - 1410, July 2013.
- **Zongyang Ma**, Aixin Sun, Gao Cong. Will This #Hashtag Be Popular Tomorrow? Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval (**SIGIR 2012**). Pages 1173-1174, Portland, OR, USA, August 12 - 16, 2012.
- Quan Yuan, Gao Cong, Kaiqi Zhao, **Zongyang Ma**, Aixin Sun. Who, Where, When and What: a Non-parametric Bayesian Approach to Context-aware Recommendation and Search for Twitter Users. *ACM Transactions on Information Systems (TOIS)*. January 2015.
- Quan Yuan, Gao Cong, **Zongyang Ma**, Aixin Sun, Nadia Magnenat-Thalmann. Who, Where, When and What: Discover Spatio-Temporal Topics for Twitter Users. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (**KDD 2013**), Pages 605-613. Chicago, IL, USA, August 11 - 14, 2013.
- Quan Yuan, Gao Cong, **Zongyang Ma**, Aixin Sun, Nadia Magnenat-Thalmann. Time-aware point-of-interest recommendation. Proceedings of the 36th international ACM SIGIR Conference on Research and Development in Information Retrieval (**SIGIR 2013**), Pages 363-372, Dublin, Ireland, July 28 - August 1, 2013.

Bibliography

- [AC10] Deepak Agarwal and Bee-Chung Chen. flda: Matrix factorization through latent dirichlet allocation. In *WSDM*, pages 91–100. ACM, 2010.
- [AH13] Xiangdong An and Jimmy Xiangji Huang. Boosting novelty for biomedical information retrieval through probabilistic latent semantic analysis. In *SIGIR*, pages 829–832. ACM, 2013.
- [AHKL12] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Discovering value from community activity on focused question answering sites: A case study of stack overflow. In *KDD*, pages 850–858. ACM, 2012.
- [AHKL13] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *WWW*, pages 95–106. International World Wide Web Conferences Steering Committee / ACM, 2013.
- [All02] James Allan, editor. *Topic Detection and Tracking*. Kluwer Academic Publishers, 2002.
- [BGJ10] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57(2):7:1–7:30, feb 2010.

- [BJ03] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR*, pages 127–134. ACM, 2003.
- [BKLDNM13] Lars Backstrom, Jon Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. Characterizing and curating conversation threads: Expansion, focus, volume, re-entry. In *WSDM*, pages 13–22. ACM, 2013.
- [BL06] David M. Blei and John D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120. ACM, 2006.
- [BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- [BSB08] István Bíró, Jácint Szabó, and András A. Benczúr. Latent dirichlet allocation in web spam filtering. In *AIRWeb*, pages 29–32. ACM, 2008.
- [BSSB09] István Bíró, Dávid Siklósi, Jácint Szabó, and András A. Benczúr. Linked latent dirichlet allocation in web spam filtering. In *AIRWeb*, pages 37–40. ACM, 2009.
- [BYT⁺09] Markus Bundschuh, Shipeng Yu, Volker Tresp, Achim Rettinger, Mathaeus Dejori, and Hans-Peter Kriegel. Hierarchical bayesian models for collaborative tagging systems. In *ICDM*, pages 728–733. IEEE Computer Society, 2009.
- [CB09] Jonathan Chang and David M. Blei. Relational topic models for document networks. In *AISTATS*, pages 81–88. JMLR.org, 2009.
- [CG98] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336. ACM, 1998.

- [CHA⁺11] Xin Chen, Xiaohua Hu, Yuan An, Zunyan Xiong, Tingting He, and E. K. Park. Perspective hierarchical dirichlet process for user-tagged image modeling. In *CIKM*, pages 1341–1346. ACM, 2011.
- [CJS11] Mengen Chen, Xiaoming Jin, and Dou Shen. Short text classification improved by learning multi-granularity topics. In *IJCAI*, pages 1776–1781. IJCAI/AAAI, 2011.
- [CKM13] James Cook, Krishnaram Kenthapadi, and Nina Mishra. Group chats on twitter. In *WWW*, pages 225–236. International World Wide Web Conferences Steering Committee / ACM, 2013.
- [CORW09] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):17:1–17:38, 2009.
- [Cow04] Philip J. Cowans. Information retrieval using hierarchical dirichlet processes. In *SIGIR*, pages 564–565. ACM, 2004.
- [CS14] Denzil Correa and Ashish Sureka. Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow. In *WWW*, pages 631–642. International World Wide Web Conferences Steering Committee / ACM, 2014.
- [CWCC14] Victor W. Chu, Raymond K. K. Wong, Fang Chen, and Chi-Hung Chi. Microblog topic contagiousness measurement and emerging outbreak monitoring. In *CIKM*, pages 1099–1108. ACM, 2014.
- [DADSTN12] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. Real-time top-n recommendation in social streams. In *RecSys*, pages 59–66. ACM, 2012.

- [DDF⁺90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JASIST*, 41(6):391–407, 1990.
- [DGCC13] Daniel Hasan Dalip, Marcos André Gonçalves, Marco Cristo, and Pavel Calado. Exploiting user feedback to learn to rank answers in Q&A forums: A case study with stack overflow. In *SIGIR*, pages 543–552. ACM, 2013.
- [DJZL12] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding bursty topics from microblogs. In *ACL*, pages 536–544. Association for Computational Linguistics, 2012.
- [DKL08] Hongbo Deng, Irwin King, and Michael R. Lyu. Formal models for expert finding on dblp bibliography data. In *ICDM*, pages 163–172. IEEE Computer Society, 2008.
- [DKMS11] Gideon Dror, Yehuda Koren, Yoelle Maarek, and Idan Szpektor. I want to answer; who has a question? Yahoo! answers recommender system. In *KDD*, pages 1109–1117. ACM, 2011.
- [DLP03] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528. ACM, 2003.
- [DM06] Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *ACL*, pages 305–312. Association for Computational Linguistics, 2006.
- [DSS12] Onkar Dalal, Srinivasan H. Sengemedu, and Subhajit Sanyal. Multi-objective ranking of comments on web. In *WWW*, pages 419–428. ACM, 2012.

- [DTR10] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *COLING*, pages 241–249. ACL, 2010.
- [FW12] Wei Feng and Jianyong Wang. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. In *KDD*, pages 1276–1284. ACM, 2012.
- [GBM⁺09] Ziyu Guan, Jiajun Bu, Qiaozhu Mei, Chun Chen, and Can Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *SIGIR*, pages 540–547. ACM, 2009.
- [GCXZ11] Jiafeng Guo, Xueqi Cheng, Gu Xu, and Xiaofei Zhu. Intent-aware query similarity. In *CIKM*, pages 259–268. ACM, 2011.
- [Gil99] W. R. Gilks. *Markov Chain Monte Carlo In Practic*. Chapman and Hall/CRC, 1999.
- [GL11] Sheng Gao and Haizhou Li. A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. In *CIKM*, pages 1047–1052. ACM, 2011.
- [GS04] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *PNAS*, 101(suppl. 1):5228–5235, 2004.
- [HAG⁺12] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsoulis. Discovering geographical topics in the twitter stream. In *WWW*, pages 769–778. ACM, 2012.
- [HBB10] Matthew Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *NIPS*, pages 856–864. Curran Associates, Inc., 2010.

- [HCC11] Lichan Hong, Gregorio Convertino, and Ed H. Chi. Language matters in twitter: A large scale study. In *ICWSM*. The AAAI Press, 2011.
- [HKC09] Chiao-Fang Hsu, Elham Khabiri, and James Caverlee. Ranking comments on the social web. In *Proc. Int’l Conf. on Computational Science and Engineering (Vol. 04)*, pages 90–97, 2009.
- [HKXC14] Xiangnan He, Min-Yen Kan, Peichu Xie, and Xiao Chen. Comment-based multi-view clustering of web 2.0 items. In *WWW*, pages 771–782. International World Wide Web Conferences Steering Committee / ACM, 2014.
- [Hof99] Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57. ACM, 1999.
- [Hof03] Thomas Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR*, pages 259–266. ACM, 2003.
- [Hof04] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):89–115, 2004.
- [HSL08] Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: Understanding documents with readers’ feedback. In *SIGIR*, pages 291–298. ACM, 2008.
- [HSZC09] Xia Hu, Nan Sun, Chao Zhang, and Tat-Seng Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *CIKM*, pages 919–928. ACM, 2009.
- [JLZ⁺11] Ou Jin, Nathan N. Liu, Kai Zhao, Yong Yu, and Qiang Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 775–784. ACM, 2011.

- [JO11] Yohan Jo and Alice H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824. ACM, 2011.
- [JS10] Ramesh Jain and Pinaki Sinha. Content without context is meaningless. In *ACM Multimedia*, pages 1259–1268. ACM, 2010.
- [KC13] Krishna Y. Kamath and James Caverlee. Spatio-temporal meme prediction: Learning what hashtags will be popular where. In *CIKM*, pages 1341–1350. ACM, 2013.
- [KCH11] Elham Khabiri, James Caverlee, and Chiao-Fang Hsu. Summarizing user-contributed comments. In *ICWSM*. The AAAI Press, 2011.
- [KFN09] Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. Latent dirichlet allocation for tag recommendation. In *RecSys*, pages 61–68. ACM, 2009.
- [KHC09] Elham Khabiri, Chiao-Fang Hsu, and James Caverlee. Analyzing and predicting community preference of socially generated metadata: A case study on comments in the digg community. In *ICWSM*. The AAAI Press, 2009.
- [KHLZ12] Su Mon Kywe, Tuan-Anh Hoang, Ee-Peng Lim, and Feida Zhu. On recommending hashtags in twitter networks. In *SocInfo*, pages 337–350. Springer-Verlag, 2012.
- [KPS13] Younghoon Kim, Yoonjae Park, and Kyuseok Shim. Digtobi: A recommendation system for digg articles using probabilistic modeling. In *WWW*, pages 691–702. International World Wide Web Conferences Steering Committee / ACM, 2013.
- [KSK12] Ravi Kant, Srinivasan H. Sengamedu, and Krishnan S. Kumar. Comment spam detection by sequence mining. In *WSDM*, pages 183–192. ACM, 2012.

- [LH09] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384. ACM, 2009.
- [LHAY07] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Arsa: A sentiment-aware model for predicting sales performance using blogs. In *SIGIR*, pages 607–614. ACM, 2007.
- [LHC⁺10] Caimei Lu, Xiaohua Hu, Xin Chen, Jung-Ran Park, TingTing He, and Zhoujun Li. The topic-perspective model for social tagging systems. In *KDD*, pages 683–692. ACM, 2010.
- [LNMG09] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: Joint models of topic and author community. In *ICML*, pages 665–672. ACM, 2009.
- [LSG12] Ze Li, Haiying Shen, and Joseph Edward Grant. Collective intelligence in the online social network of Yahoo! answers and its implications. In *CIKM*, pages 455–464. ACM, 2012.
- [LXTC12] Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. Time-aware topic recommendation based on micro-blogs. In *CIKM*, pages 1657–1661. ACM, 2012.
- [LZMH10] Cindy Xide Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. Pet: A statistical model for popular events tracking in social communities. In *KDD*, pages 929–938. ACM, 2010.
- [MCZZ08] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110. ACM, 2008.
- [MG06] Gilad Mishne and Natalie Glance. Leave a reply: An analysis of weblog comments. In *In Third annual workshop on the Weblogging ecosystem*, 2006.

- [MJ] Allie Mazzia and James Juett. Suggesting hashtags on twitter. <http://www-personal.umich.edu/~amazzia/>.
- [MLSZ06] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW*, pages 533–542. ACM, 2006.
- [MLW⁺07] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *WWW*, pages 171–180. ACM, 2007.
- [MNBD06] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *ACM HyperText’06*, pages 31–40. ACM, 2006.
- [MZ05] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: An exploration of temporal text mining. In *KDD*, pages 198–207. ACM, 2005.
- [NASW09] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10:1801–1828, dec 2009.
- [NBG11] Mor Naaman, Hila Becker, and Luis Gravano. Hip and trendy: Characterizing emerging trends on twitter. *J. Am. Soc. Inf. Sci. Technol.*, 62(5):902–918, 2011.
- [NCD⁺07] Ramesh Nallapati, William Cohen, Susan Dittmore, John Lafferty, and Kin Ung. Multi-scale topic tomography. In *KDD*, pages 520–529. ACM, 2007.
- [NH99] Radford M. Neal and Geoffrey E. Hinton. *Learning in Graphical Models*. MIT Press, 1999.

- [PC98] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR*, pages 275–281. ACM, 1998.
- [PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: the Art of Scientific Computing*. Cambridge University Press, 1988.
- [PNH08] Xuan Hieu Phan, Minh Le Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, pages 91–100. ACM, 2008.
- [Rad13] RadiumOne. #mobile hashtagsurvey, 2013. <http://radiumone.com/about/company-resources.html>.
- [RBMNST09] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *KDD*, pages 727–736. ACM, 2009.
- [RMK11] Daniel M. Romero, Brendan Meeder, and Jon Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *WWW*, pages 695–704. ACM, 2011.
- [RST10] Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, pages 81–90. ACM, 2010.
- [RZGSS04] Michal Rosen-Zvi, Tom Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494. AUAI Press, 2004.
- [SCFS12] Mrinmaya Sachan, Danish Contractor, Tanveer A. Faruque, and L. Venkata Subramaniam. Using content and interactions for dis-

- covering communities in social networks. In *WWW*, pages 331–340. ACM, 2012.
- [SCNSP10] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments? analyzing and predicting youtube comments and comment ratings. In *WWW*, pages 891–900. ACM, 2010.
- [SFMC12] Ugo Scaiella, Paolo Ferragina, Andrea Marino, and Massimiliano Ciaramita. Topical clustering of search results. In *WSDM*, pages 223–232. ACM, 2012.
- [SH06] Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *WWW*, pages 377–386. ACM, 2006.
- [SHGY09] Yizhou Sun, Jiawei Han, Jing Gao, and Yintao Yu. itopicmodel: Information network-integrated topic modeling. In *ICDM*, pages 493–502. IEEE Computer Society, 2009.
- [Siz10] Sergej Sizov. Geofolk: Latent spatial semantics in web 2.0 social media. In *WSDM*, pages 281–290. ACM, 2010.
- [SKKL12] Erez Shmueli, Amit Kagian, Yehuda Koren, and Ronny Lempel. Care to comment? recommendations for commenting on news stories. In *WWW*, pages 429–438. ACM, 2012.
- [SLH14] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, 2014.
- [SM07] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264. Curran Associates, Inc., 2007.

- [SMdR07] Anne Schuth, Maarten Marx, and Maarten de Rijke. Extracting the discussion structure in comments on news-articles. In *WSDM*, pages 97–104. ACM, 2007.
- [SNM08] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Tag recommendations based on tensor dimensionality reduction. In *RecSys*, pages 43–50. ACM, 2008.
- [SPUP02] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *SIGIR*, pages 253–260. ACM, 2002.
- [SPYO12] Jose San Pedro, Tom Yeh, and Nuria Oliver. Leveraging user comments for aesthetic aware image search reranking. In *WWW*, pages 439–448. ACM, 2012.
- [SS14] Surendra Sedhai and Aixin Sun. Hashtag recommendation for hyper-linked tweets. In *SIGIR*, pages 831–834. ACM, 2014.
- [SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.
- [TJBB06] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *JASA*, 101(476):1566–1581, 2006.
- [TM08] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008.
- [TMN⁺14] Jian Tang, Zhaoshi Meng, XuanLong Nguyen, Qiaozhu Mei, and Ming Zhang. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *ICML*, pages 190–198. ACM, 2014.

- [TR12] Oren Tsur and Ari Rappoport. What’s in a hashtag? content based prediction of the spread of ideas in microblogging communities. In *WSDM*, pages 643–652. ACM, 2012.
- [TWdR09] Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. Predicting the volume of comments on online news stories. In *CIKM*, pages 1765–1768. ACM, 2009.
- [TYC09] Jie Tang, Limin Yao, and Dewei Chen. Multi-topic based query-oriented summarization. In *SDM*, pages 1147–1158. SIAM, 2009.
- [Wal06] Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *ICML*, pages 977–984. ACM, 2006.
- [WB11] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *KDD*, pages 448–456. ACM, 2011.
- [WBH08] Chong Wang, David M. Blei, and David Heckerman. Continuous time dynamic topic models. In *UAI*, pages 579–586. AUAI Press, 2008.
- [WL04] Tak-Lam Wong and Wai Lam. A probabilistic approach for adapting information extraction wrappers and discovering new attributes. In *ICDM*, pages 257–264. IEEE Computer Society, 2004.
- [WLZ10] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: A rating regression approach. In *KDD*, pages 783–792. ACM, 2010.
- [WMW07] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *ICDM*, pages 697–702. IEEE Computer Society, 2007.
- [WSW07] Xing Wei, Jimeng Sun, and Xuerui Wang. Dynamic mixture models for multiple time series. In *IJCAI*, pages 2909–2914. Morgan Kaufmann Publishers Inc., 2007.

- [WWL⁺11] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach. In *CIKM*, pages 1031–1040. ACM, 2011.
- [WZY06] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW*, pages 417–426. ACM, 2006.
- [XJW12] Fei Xu, Zongcheng Ji, and Bin Wang. Dual role model for question recommendation in community question answering. In *SIGIR*, pages 771–780. ACM, 2012.
- [YCH⁺11] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In *WWW*, pages 247–256. ACM, 2011.
- [YCM⁺13] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Who, where, when and what: Discover spatio-temporal topics for twitter users. In *KDD*, pages 605–613. ACM, 2013.
- [YCZ⁺15] Quan Yuan, Gao Cong, Kaiqi Zhao, Zongyang Ma, and Aixin Sun. Who, where, when and what: a non-parametric bayesian approach to context-aware recommendation and search for twitter users. *TOIS*, 2015.
- [YGC⁺13] Dawei Yin, Shengbo Guo, Boris Chidlovskii, Brian D. Davison, Cedric Archambeau, and Guillaume Bouchard. Connecting comments and tags: Improved modeling of social tagging systems. In *WSDM*, pages 547–556. ACM, 2013.
- [YJCZ09] Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: A discriminative approach. In *KDD*, pages 927–936. ACM, 2009.

- [YLHA12] Xiaohui Yu, Yang Liu, Xiangji Huang, and Aijun An. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Trans. Knowl. Data Eng.*, 24(4):720–734, 2012.
- [YQG⁺13] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. CQArank: Jointly model topics and expertise in community question answering. In *CIKM*, pages 99–108. ACM, 2013.
- [YSZM12] Lei Yang, Tao Sun, Ming Zhang, and Qiaozhu Mei. We know what @you #tag: Does the dual role affect hashtag adoption? In *WWW*, pages 261–270. ACM, 2012.
- [YYLF09] W. G. Yee, A. Yates, S. Liu, and O. Frieder. Are web user comments useful for search. In *Workshop on Large-Scale Distributed Systems for Information Retrieval (LSDS-IR) at SIGIR’09*, pages 63–70, 2009.
- [ZGS11] Eva Zangerle, Wolfgang Gassler, and Günther Specht. Recommending #-tags in twitter. In *SASWeb*, pages 67–78. CEUR-WS.org, 2011.
- [ZJW⁺11] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *ECIR*, pages 338–349. Springer-Verlag, 2011.
- [ZML⁺06] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *WWW*, pages 173–182. ACM, 2006.
- [ZSZL10] Jianwen Zhang, Yangqiu Song, Changshui Zhang, and Shixia Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *KDD*, pages 1079–1088. ACM, 2010.

- [ZZS⁺14] Ying Zhang, Ning Zhang, Luo Si, Yanshan Lu, Qifan Wang, and Xiaojie Yuan. Cross-domain and cross-category emotion tagging for comments of online news. In *SIGIR*, pages 627–636. ACM, 2014.
- [ZZSW09] Huibin Zhang, Mingjie Zhu, Shuming Shi, and Ji-Rong Wen. Employing topic models for pattern-based semantic class discovery. In *ACL*, pages 459–467. Association for Computational Linguistics, 2009.