Learning with Social Media

ZHOU, Chao

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
April 2013

Thesis/Assessment Committee

Professor YU Xu Jeffrey (Chair)

Professor LYU Rung Tsong Michael (Thesis Supervisor)

Professor KING Kuo Chin Irwin (Thesis Supervisor)

Professor ZHANG Shengyu (Committee Member)

Professor YANG Qiang (External Examiner)

Abstract of thesis entitled:
Learning with Social Media
Submitted by ZHOU, Chao
for the degree of Doctor of Philosophy
at The Chinese University of Hong Kong in April 2013

With the astronomical growth of Web 2.0 over the past decade, social media systems, such as rating systems, social tagging systems, online forums, and community-based question answering (Q&A) systems, have revolutionized people's way of creating and sharing contents on the Web. However, due to the explosive growth of data in social media systems, users are drowning in information and encountering information overload problem. Currently, social computing techniques, achieved through learning with social media, have emerged as an important research area to help social media users find their information needs. In general, users post contents which reflect their interests in social media systems, and expect to obtain the suitable items through social computing techniques. To better understand users' interests, it is very essential to analyze different types of user generate content. On the other hand, the returned information may be items, or users with similar interests. Beyond the user-based analysis, it would be quite interesting and important to conduct item-oriented study, such as understand items' characteristics, and grouping items that are semantically related for better addressing users' information needs.

The objective of this thesis is to establish automatic and scalable models to help social media users find their information needs more effectively. These models are proposed based on the two key entities in social media systems: user and item. Thus, one important aspect of this thesis is therefore to develop a framework to combine the user information and the item information with the following two purposes: 1) modeling users' interests with respect to their behavior, and recommending items or users they may be interested in; and 2) understanding items' characteristics, and grouping items that are semantically related for better addressing users' information needs.

For the first purpose, a novel unified matrix factorization framework which fuses different types of users' behavior data, is proposed for predicting users' interests on new items. The framework tackles the data sparsity problem and non-flexibility problem confronted by traditional algorithms. Furthermore, to provide users with an automatic and effective way to discover other users with common interests, we propose a framework for user interest modeling and interest-based user recommendation by utilizing users' tagging information. Extensive evaluations on real world data demonstrate the effectiveness of the proposed user-based models.

For the second purpose, a new functionality question suggestion, which targets at suggesting questions that are semantically related to a queried question, is proposed in social media systems with Q&A functionalities. Existing bag-of-words approaches suffer from the shortcoming that they could not bridge the lexical chasm between semantically related questions. Therefore, we present two models which combines both the lexical and latent semantic knowledge to measure the semantic relatedness among questions. In question analysis, there is a lack of understanding of questions' characteristics. To tackle this problem, a supervised approach is developed to identify questions' subjectivity. Moreover, we come up with an approach to collect training data automatically by utilizing social signals without involving any manual labeling. The experimental results show that our methods perform better than the state-of-the-art approaches.

In summary, based on the two key entities in social media systems, we present two user-based models and two item-oriented models to help social media users find their information needs more accurately and effectively through learning with social media. Extensive experiments on various social media systems confirm the effectiveness of proposed models.

論文題目 ： 基於社會化媒體的學習

作者 ： 周超

學校 ： 香港中文大學

學系 ： 計算機科學及工程學系

修讀學位 ： 哲學博士

摘要 ：

隨著Web 2.0系統在過去十年的迅猛發展，社會化媒體，比如社會化評分系統、社會化標籤系統、在線論壇和社會化問答系統，已經革命性地改變了人們在互聯網上創造和分享內容的方式。但是，面對社會化媒體數據的飛速增長，用戶面臨嚴重的信息過載的問題。現在，基於社會化媒體學習的社會化計算，已經發展成爲了幫助社會化媒體用戶有效解決信息需求的一個重要的研究領域。一般來說，用戶在社會化媒體中發佈信息，期望通過社會化計算尋找到合適的項目。爲了更好地理解用戶的興趣，分析不同類型的用戶產生數據是非常重要的。另一方面，返回給用戶的可以是項目，或是擁有相似興趣的其他用戶。除了基於用戶的分析，進行基於項目的分析也是非常有趣和重要的，比如理解項目的屬性，將語義相關的項目聚在一起爲了更好地滿足用戶的信息需求等。

本論文的目地是提出自動化和可擴展的模型來幫助社會化媒體用戶更有效的解決信息需求。這些模型基於社會化媒體中兩個重要的組成提出：用戶和項目。因此，基於以下兩個目標，我們提出一個統一的框架來整合用戶信息和項目信息：1)通過用戶的行為找出用戶的興趣，并為之推薦可能感興趣的項目和相似興趣的用戶；2)理解項目的屬性，並將語義相關的項目聚合在一起從而能更好的滿足用戶的信息需求。

爲了完成第一個目標，我們提出了一個新的矩陣分解的框架來整合不同的用戶行為數據，從而預測用戶對新項目的興趣。這個框

架有效地解決了數據稀疏性以及傳統方法中信息來源單一的問題，其次，爲了給社會化媒體用戶提供自動發現類似興趣的其他用戶的方式，通過利用社會化標籤信息，我們提出了基於用戶興趣挖掘和基於興趣的用戶推薦的框架。大量的真實數據實驗驗證了提出的基於用戶的模型的有效性。

爲了完成第二個目標，我們在具問答性質的社會化媒體中提出了問題推薦的應用。問題推薦的目標是基於一個用戶問題推薦語義相關的問題。傳統的詞袋模型不能有效地解決相關問題中用詞不同的問題。因此，我們提出了兩個模型來結合詞法分析以及潛在語義分析，從而有效地衡量問題間的語義相關度。在問題分析中，當前研究缺少對問題屬性的認識。爲了解決這個問題，我們提出了一個有監督學習的方法來識別問題的主觀性。具體來說，我們提出了一種基於社會化信號的無人工參與的自動收集訓練數據的方法。大量實驗證實了提出的方法的效果超過了之前的其他算法。

概括起來，圍繞社會化媒體中兩個重要的組成，我們提出了兩個基於用戶的模型和兩個基於項目的模型來幫助社會化媒體的用戶更準確更有效地解決信息需求。我們通過不同社會化媒體中的大量實驗證實了提出模型的有效性。

Acknowledgement

I would like to express my sincere gratitude and appreciation to my supervisors, Prof. Irwin King and Prof. Michael R. Lyu. I gain too much from their guidance not only on knowledge and attitude in doing research, but also on the presentation, teaching, and English writing skills. I will always be grateful for their supervision, encouragement and support at all levels.

I am grateful to my thesis committee members, Prof. Jeffrey Xu Yu and Prof. Shengyu Zhang for their helpful comments and suggestions about this thesis. My special thanks to Prof. Qiang Yang who kindly served as the external committee for this thesis.

I would like to thank my mentors, Dr. Edward Y. Chang in Google, Dr. Chao Liu and Dr. Yi-Min Wang in Microsoft Research Redmond, Dr. Tai-Yi Huang in Microsoft Bing, Dr. Tat-Seng Chua in National University of Singapore, Dr. Chin-Yew Lin, Dr. Young-In Song and Dr. Yunbo Cao in Microsoft Research Asia, for their guidance, support, insightful opinions and valuable suggestions when I was visiting as a research intern.

I thank Hao Ma, Hongbo Deng, Zenglin Xu, Haiqin Yang, Kaizhu Huang, Jianke Zhu, Xin Xin and Zibin Zheng for their effort and constructive discussions in conducting the research work in this thesis. I also thank my colleagues in the web intelligence and social computing group, Allen Lin, Baichuan Li, Shenglin Zhao, Hongyi Zhang, Priyanka Garg, Shouyuan Chen, Chen Cheng, Guang Ling. I also appreciate the help from my officemates and friends, Qirun Zhang, Xinyu Chen, Xiaoqi Li, Yangfan Zhou, Wujie Zheng, Junjie

Xiong, Jieming Zhu, Yilei Zhang, Ning Wang and many others. I also would like to thank Xinying Song, Jingtian Jiang, Xiance Si, Fangtao Li, Hongji Bao, Decheng Dai, Kai Wang, Zhen Liao, and many others for their discussions in my internships.

Last but not least, I would like to thank my wife, my parents, and my parents-in-law. Without their deep love and constant support, this thesis would never have been completed.

To my beloved parents and my wife Jie.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

With the inception of Web 2.0 in World Wide Web, huge amount of User Generate Content (UGC) has been aggregated in social media systems. There are many types of social media systems that enable users to perform different tasks. Movie rating systems, such as Netflix[1], MovieLens[2], and Douban[3], allow users to rate movies. In return, an active user would receive suggested movies that he/she may be interested in based on rating information of other users or items [133, 262]. The rated information items could also be books, music, news, Web pages, products, etc. Social tagging systems, such as Delicious[4], Flickr[5], and CiteULike[6], have emerged as an effective way for users to annotate and share objects on the Web. Tags posted by users on bookmarks, papers, and photos, express users' understandings and interests [263, 259]. An online forum is a Web application which involves highly interactive and semantically related discussions on domain specific questions, such as travel, sports, and programming [260]. Community-based Q&A services, such as Ya-

[1] http://www.netflix.com

[2] http://movielens.umn.edu

[3] http://movie.douban.com/

[4] https://delicious.com/

[5] http://www.flickr.com/

[6] http://www.citeulike.org/

Table 1.1: Overview of work in thesis.

Work	**Source**	**Goal**
Item Recommendation with Tagging Ensemble	User	Item
User Recommendation via Interest Modeling	User	User
Item Suggestion with Semantic Analysis	Item	Item
Item Modeling via Data-Driven Approach	Item	Characteristic of Item

hoo! Answers[7], Baidu Knows[8], and Quora[9], are online communities that adopt the Web 2.0 model and organize knowledge exchange in the form of asking and answering questions [264, 261].

Although social media systems are designed for different purposes, there are two key entities in each social media system: user and item. Users, as the participants of social media systems, post materials, browse contents, interact with other users, and discovery information. Items, generated either by a social media system or users, act as the consumption entities in the ecosystem of Web 2.0.

With its astronomical growth over the past decade, the data in social media systems become huge, diverse and dynamic. As a result, users are currently drowning in information and facing information overload [135]. *To help social media users find their information need, a critical issue is to model users' interests with respect to their behavior, and recommend items or users they may be interested in. On the other hand, it would be quite interesting and important to understand items' characteristics, and group items that are semantically related for better addressing users' information needs.* In order to achieve above goals, in this thesis, we present four studies on learning with social media from different perspectives. Table 1.1 shows the overview of work in thesis.

In Table 1.1, *source* means *who to recommend*, *goal* means *what to recommend*.

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

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

[9] http://www.quora.com

In the first work *Item Recommendation with Tagging Ensemble*, we target at recommending items to a given user based on information from rating systems. With the development of social media systems, huge amount of User Generate Content (UGC) is generated each day. Users are easily overwhelmed by the rapidly aggregated information in social media systems. Solving the information overload problem by providing users with more proactive and personalized information has becoming increasingly indispensable nowadays. Thus, recommender systems research has become an important research area aiming at tackling the information overload problem [78, 181, 206].

Typically, recommender system is based on collaborative filtering. The user-item rating matrix is usually constructed in collaborative filtering. The target of collaborative filtering is to fill the rating matrix according to existing observed ratings. Two types of collaborative filtering approaches are widely studied: memory-based [30, 51, 63] and model-based [26, 35].

Memory-based collaborative filtering approaches are widely investigated [89, 98, 115] and employed in industrial collaborative filtering systems [115, 181, 49]. The most studied memory-based approaches include user-based approaches [30, 75, 89] and item-based approaches [200, 53, 115]. User-based methods look for some similar users who have similar rating styles with the active user and then employ the ratings from those similar users to predict the ratings for the active user [206]. Item-based approaches share similar idea with user-based methods except for predicting the ratings of active users based on the information of similar items computed [53, 200].

Different from memory-based collaborative filtering, model-based approaches first train a model based on observed user-item ratings, and then employ the trained model to predict missing values [26, 35]. Many model-based approaches are proposed, include aspect models [80, 208], Bayesian model [42], relevance models [230, 232], latent class models [81, 91, 138], matrix factorization models [26, 64], and clustering models [12, 60, 97].

Although collaborative filtering algorithms have been widely used in recommendation systems, the problem of inaccurate recommendation results still exists in both neighborhood-based methods and model-based methods. The fundamental problem of these approaches is the data sparsity of the user-item rating matrix. The density of available ratings in commercial recommender systems is often less than 1% [115] or even much less. Thus, only utilizing user-item rating information as most collaborative filtering algorithms do is not enough. Social tagging systems have recently emerged as a popular ways for users to annotate, organize, and share resources on the Web. Previous studies [76, 112, 202] have shown that tags can represent users' judgments about Web contents quite accurately, which are also good candidates to describe the resources. To overcome the data sparsity problem and non-flexibility problem confronted by traditional collaborative filtering algorithms, we propose a factor analysis approach by utilizing both users' rating information and tagging information based on probabilistic matrix factorization in the first study.

In the second work *User Recommendation via Interest Modeling*, we focus on recommending users to a given user with similar interests in social tagging systems. Social tagging systems have emerged as a popular way for users to annotate, organize, and share resources on the Web, such as Yahoo! Delicious, Flickr and CiteULike. Social tagging systems enjoy the advantages that users can use free-form tags to annotate objects, which can ease sharing of objects despite vocabulary differences. As a form of users' individual behavior, tagging activity not only can represent users' judgments on the resources [76, 210], but also can indicate users' personal interests [220]. However, due to the fast growth of social tagging systems, a user is easily overwhelmed by the large amount of data and it is very difficult for the user to dig out information that he/she is interested in. Several functions aiming at finding people with similar interests have been incorporated into tagging systems, such as *network* in Yahoo! Delicious and *contact* in Flickr. Take *network* in Yahoo! Delicious as an example, if a user Bob notices many of Jack's bookmarks as interesting, Bob can add Jack to his *network*. After that, when Jack updates his new bookmarks, they will also appear in Bob's bookmark pool to make it more convenient for Bob to browse resources he is interested in. However, no automatic interest-based user recommendation service is provided and it is not easy for a user to find other users with similar interest.

Thus, in the second study, we propose an effective two-phase *User Recommendation* (*UserRec*) framework for users' interest modeling and interest-based user recommendation, which can help information sharing among users with similar interests. Solving the problem of modeling users' interests and performing interest-based user recommendation in social tagging systems achieve two benefits. At a fundamental level, we gain insights into utilizing information in social tagging systems to provide personalized service for each user. At a practical level, it can bring several enhancements. Firstly, it is more convenient for an active user to know the latest resources on particular topics he/she may be interested in because users with similar interests are recommended. Secondly, it can help users obtain high-quality results through social filtering. Thirdly, interest-based user recommendation can help build interest-based social relationships, and forming interest-based social groups, therefore increasing intra-group information flow on the corresponding topics.

In the third work *Item Suggestion with Semantic Analysis*, we propose an approach to recommend items to a given item that are semantically related in online forums and community-based Q&A systems. This study is related to automatic Q&A, which has been a long-standing research problem that attracts contributions from the information retrieval and natural language processing communities. Automatic Q&A ranges from automatic subjective Q&A [109, 215] to automatic factual Q&A [52, 56, 70].

Most work of retrieving answers directly from the Web focus on factual Q&A [178, 177, 114, 205, 3]. However, the existing methods for automatic Q&A from the Web do not utilize readily available Q&A pairs in social media as they just extract answers directly from the Web for questions. With the popularization of social media with Q&A aspect, people has come together to post their questions, answer other users' questions, and interact with each other. Community-based Q&A services and online forums are two representative platforms for this purpose. Overtimes, a large amount of historical Q&A pairs have been built up in their archives, providing information seekers a viable alternative to Q&A from the Web [1, 44, 84].

With the proliferation of community-based Q&A services and online forums, a large amount of historical Q&A pairs have been accumulated in social media systems. Researchers have been investigating automatic question answering in social media systems recently. Question search aims at finding semantically equivalent questions for a user question. Addressing the lexical chasm problem between user questions and the questions in a Q&A archive is the focus of most existing work. Berger et al. [20] studied four statistical techniques for bridging the lexical chasm, which include adaptive TFIDF [179], automatic query expansion [147], statistical translation models [21], and latent semantic models [79]. The history of question search originated from FAQ retrieval. The FAQ Finder combined lexical similarity and semantic similarity between questions to rank FAQs, where a vector space model was employed to compute the lexical similarity and the WordNet [145] was utilized to capture the semantic similarity [34]. Recently, question search has been re-visited with the Q&A data in social media, which mainly includes community-based Q&A services and online forums. Jeon et al. [87, 86] employed translation model to tackle the question search problem in community-based Q&A. Translation model, proposed by Berger et al. [21], has been extensively employed in question finding and answer retrieval [20, 54, 88, 184]. However, most existing methods of question search only find equivalent questions instead of semantically related questions. In addition, most existing methods only utilize lexical information.

Thus, in the third study, we propose a new function for Q&A in social media, named question suggestion, a functionality facilitating a user to explore a topic he/she is interested in by suggesting semantically related questions to a queried question. Performing question suggestion in social media systems has three benefits: (1) helping users explore their information needs thoroughly from different perspectives; (2) increasing page views by enticing users' clicks on suggested questions to increase potential revenues; (3) providing social media systems a relevance feedback mechanism by mining users' click through logs to improve search quality. We present a framework to suggest questions, and propose the Topic-enhanced Translation-based Language Model (TopicTRLM) which fuses both the lexical and latent semantic knowledge. Moreover, to incorporate the answer information into the model to make the model more complete, we also propose the Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A).

In the fourth work *Item Modeling via Data-Driven Approach*, we aim at finding the characteristic for an item in community-based Q&A systems. Automatic Question Answering (AQA) has been a long-standing research problem which attracts contributions from the information retrieval and natural language processing communities. AQA ranges from Automatic Subjective Question Answering (ASQA) [215, 110] to Automatic Factual Question Answering (AFQA) [70, 52, 56]. Although much progress has been made in AFQA, with the notable example of the IBM Watson system [56], high quality ASQA is still beyond the state-of-the-art. There are two fundamental differences of ASQA compared with AFQA: firstly, ASQA aims at returning opinions instead of facts; secondly, ASQA aims at returning an answer summarized from different perspectives instead of a fixed answer.

The rising and popularity of Community Question Answering (CQA) sites provides an alternative to ASQA. CQA sites such as Yahoo! Answers[10], Google Confucius [209], and Baidu Knows[11] provide platforms for people to post questions, answer questions, and give feedbacks to the posted items [1, 123]. The structure of QA archives from CQA sites makes these QA pairs extremely valuable to ASQA [244, 260, 261]. However, the inherently ill-phrased, vague, and complex nature of questions in CQA sites makes question analysis challenging. In addition, the lack of labeled data hinders the adventure of effective question analysis.

The explicit support of social signals in CQA sites, such as rating content, voting answers, and posting comments, aggregates rich knowledge of community wisdom. Thus, it is worthwhile to investigate whether we can leverage these social signals to advance question analysis. Motivated by Halevy, Norvig and Pereira's argument "Web-scale learning is to use available large-scale data rather than hoping for annotated data that isn't available" [69], and inspired by the unreasonable effectiveness of data in statistical speech recognition, statistical machine translation [69], and semantic relationship learning [183], our approach works towards utilizing social signals to collect training data for question analysis without manual labeling.

Thus, in the fourth study, we focus on one important aspect of question analysis: *question subjectivity identification (QSI)*. The goal is to identify whether a question is a subjective question. The asker of a subjective question expects one or more subjective answers, and the user intent is to collect people's opinions. The asker of an objective question expects an authoritative answer based on common knowledge or universal truth [4]. High quality QSI could be used to decide whether the system should try to identify the correct answer

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

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

(AFQA) or summarize a diversity of opinions (ASQA).

In summary, the first and the second studies focus on recommendation purpose, and the third and the fourth studies mainly discuss question answering related fields.

1.2 Thesis Contribution

The main contributions of this thesis could be described as follows:

1. **Item Recommendation with Tagging Ensemble**
   In order to overcome the data sparsity problem and non-flexibility problem confronted by traditional collaborative filtering algorithms, we propose a factor analysis approach, referred to as *TagRec*, by utilizing both users' rating information and tagging information based on probabilistic matrix factorization. Specifically, user-item rating matrix, user-tag tagging matrix, and item-tag tagging matrix are fused together in a unified matrix factorization framework. The experimental results on MovieLens $10M/100K$ data set show that our method performs better than the state-of-the-art approaches; in the meanwhile, our complexity analysis also implies that our approach can be scaled to very large data sets.

2. **User Recommendation via Interest Modeling**
   In order to provide users with an automatic and effective way to discover other users with common interests in social tagging systems, we propose the *User Recommendation (UserRec)* framework for user interest modeling and interest-based user recommendation, aiming to boost information sharing among users with similar interests. Firstly, we propose a tag-graph based community detection method to model the users' personal interests, which are further represented by discrete topic distributions. Secondly, the similarity values between users' topic distributions are measured by Kullback-Leibler divergence

(KL-divergence), and the similarity values are further used to perform interest-based user recommendation. Thirdly, by analyzing users' roles in a tagging system, we find users' roles in a tagging system are similar to Web pages in the Internet. Experiments on Delicious tagging data set show that UserRec outperforms other state-of-the-art recommender system approaches.

3. **Item Suggestion with Semantic Analysis**
   We propose a new functionality *Question Suggestion* in social media systems with Q&A functionalities. Question suggestion targets at suggesting questions that are semantically related to a queried question. Existing bag-of-words approaches suffer from the shortcoming that they could not bridge the lexical chasm between semantically related questions. Therefore, we present a new framework to suggest questions, and propose the *Topic-enhanced Translation-based Language Model (TopicTRLM)* which fuses both the lexical and latent semantic knowledge. Moreover, to incorporate the answer information into the model to make the model more complete, we also propose the *Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A)*. Extensive experiments have been conducted with real world data sets from a popular online forum TripAdvisor and a well known community-based Q&A service Yahoo! Answers. Experimental results indicate our approach is very effective and outperforms other popular methods in several metrics.

4. **Item Modeling via Data-Driven Approach**
   To improve the performance of question subjectivity identification in community-based Q&A services with the constrain that little labeled training data are available, we propose an approach to collect training data automatically by utilizing social signals in community-based Q&A sites without involving any manual labeling. Experimental results show that our data-

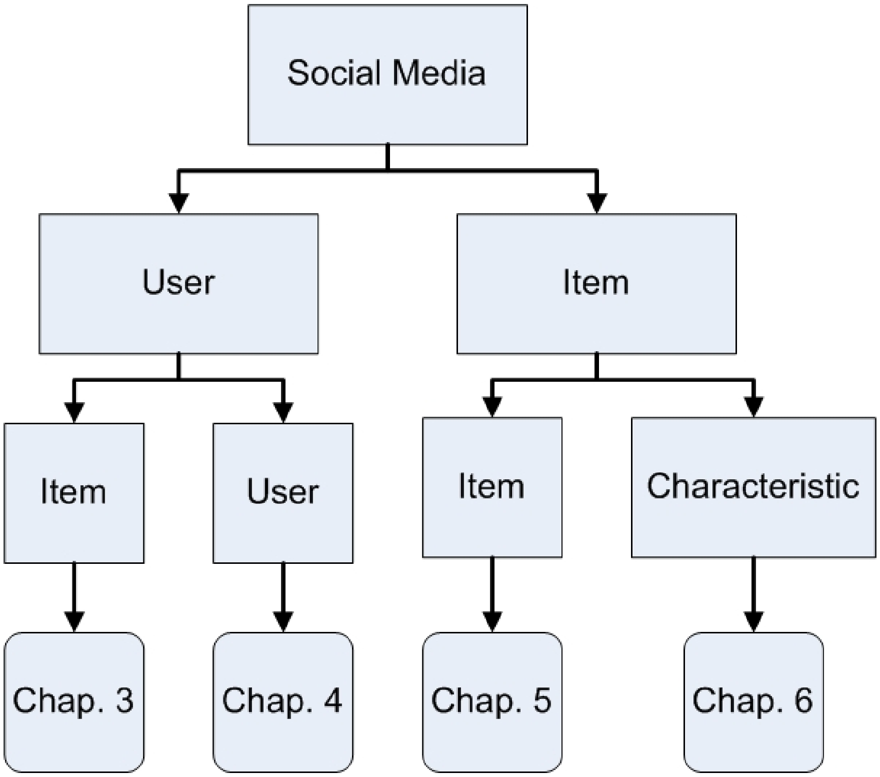


Figure 1.1: Structure of thesis contribution.

driven approach achieves 9.37% relative improvement over the supervised approach using manually labeled data, and achieves 5.15% relative gain over a state-of-the-art semi-supervised approach. In addition, we propose several heuristic features for question subjectivity identification. By adding these features, we achieve 11.23% relative improvement over word n-gram feature under the same experimental setting.

Figure 1.1 summarizes the structure of thesis contributions in each chapter.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

- **Chapter 2**
  In this chapter, we review background knowledge and related work in the field of recommender systems, machine learning, and information retrieval. We also review related applications of rating prediction, user recommendation, and automatic question answering.

- **Chapter 3**
  In this chapter, we propose a factor analysis approach, referred to as *TagRec*, by utilizing both users' rating information and tagging information based on probabilistic matrix factorization, with the target to overcome the data sparsity problem and non-flexibility problem confronted by traditional collaborative filtering algorithms [262]. Specifically, user-item rating matrix, user-tag tagging matrix, and item-tag tagging matrix are fused together in a unified matrix factorization framework. The experimental results on MovieLens $10M/100K$ data set show that our method performs better than the state-of-the-art approaches; in the meanwhile, our complexity analysis also implies that our approach can be scaled to very large data sets.

- **Chapter 4**
  This chapter focuses on providing users with an automatic and effective way to discover other users with common interests in social tagging systems. Specifically, we propose the *User Recommendation (UserRec)* framework for user interest modeling and interest-based user recommendation, aiming to boost information sharing among users with similar interests [263]. Firstly, we propose a tag-graph based community detection method to model the users' personal interests, which are further represented by discrete topic distributions. Secondly, the similarity values between users' topic distributions are measured by Kullback-Leibler divergence (KL-divergence), and the similarity values are further used to perform interest-based user rec-

ommendation. Thirdly, by analyzing users' roles in a tagging system, we find users' roles in a tagging system are similar to Web pages in the Internet. Experiments on Delicious tagging data set show that UserRec outperforms other state-of-the-art recommender system approaches.

- **Chapter 5**
  In this chapter, we propose a new functionality *Question Suggestion* in social media systems with Q&A functionalities [260]. Question suggestion targets at suggesting questions that are semantically related to a queried question. Existing bag-of-words approaches suffer from the shortcoming that they could not bridge the lexical chasm between semantically related questions. Therefore, we present a new framework to suggest questions, and propose the *Topic-enhanced Translation-based Language Model (TopicTRLM)* which fuses both the lexical and latent semantic knowledge. Moreover, to incorporate the answer information into the model to make the model more complete, we also propose the *Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A)*. Extensive experiments have been conducted with real world data sets from a popular online forum TripAdvisor and a well known community-based Q&A service Yahoo! Answers. Experimental results indicate our approach is very effective and outperforms other popular methods in several metrics.

- **Chapter 6**
  This chapter focuses on improving the performance of question subjectivity identification in community-based Q&A services with the constrain that little labeled training data are available. Specifically, we propose an approach to collect training data automatically by utilizing social signals in community-based Q&A sites without involving any manual labeling [264]. Experimental results show that our data-driven approach achieves

9.37% relative improvement over the supervised approach using manually labeled data, and achieves 5.15% relative gain over a state-of-the-art semi-supervised approach. In addition, we propose several heuristic features for question subjectivity identification. By adding these features, we achieve 11.23% relative improvement over word n-gram feature under the same experimental setting.

- **Chapter 7**
  The last chapter summarizes this thesis and addresses some future directions that can be further explored.

In order to make each of these chapters self-contained, some critical contents, e.g., model definitions or motivations having appeared in previous chapters, may be briefly reiterated in some chapters.

□ **End of chapter.**

Chapter 2

Background Review

In this chapter, we investigate techniques related to recommender systems, machine learning, and information retrieval models. Table 2.1 shows techniques studied in each work.

In addition to the studies of techniques in recommender systems, machine learning, and information retreival models, this chapter also investigate these techniques and algorithms with applications to real-world problems. Table 2.2 summarizes applications involved.

2.1 Recommender System Techniques

With the development of social media systems, huge amount of User Generate Content (UGC) is generated each day [257, 259]. Users are easily overwhelmed by the rapidly aggregated information in social media systems. Solving the information overload problem by providing users with more proactive and personalized informa-

Table 2.1: Techniques employed in the thesis.

Work	Techniques
Item Recommendation with Social Tagging Ensemble	RS
User Recommendation via Interest Modeling	RS + IR
Item Suggestion with Semantic Analysis	IR + ML
Item Modeling via Data-Driven Approach	ML

Table 2.2: Applications studied in the thesis.

Work	Applications
Item Recommendation with Tagging Ensemble	Rating prediction
User Recommendation via Interest Modeling	User recommendation
Item Suggestion with Semantic Analysis	Automatic Q&A
Item Modeling via Data-Driven Approach	Automatic Q&A

tion has becoming increasingly indispensable nowadays. Thus, recommender systems research has become an important research area aiming at tackling the information overload problem [78, 181, 206]. Authors in [2] reported that recommender systems research has connections to forecasting theories [8], information retrieval [195], cognitive science [182], approximation theory [172], management science [150] and consumer choice modeling in marketing [113]. In this section, we briefly review techniques in recommender systems. Specifically, content-based filtering and collaborative filtering. In content-based filtering, the active user would be recommended with items similar to the ones the user liked in the past. In collaborative filtering, the active user would be recommended with items that people with similar tastes and interests liked in the past.

2.1.1 Content-based Filtering

The content-based filtering adopts ideas and employs many techniques from information retrieval [195, 11, 136] and information filtering [16] research. The improvement over the traditional information retrieval approaches comes from the use of user profiles that contain information about users' tastes, preferences and needs. The profiling information can be elicited from users explicitly, e.g., through questionnaires, or implicitly learned from their behavior over time.

Content-based algorithms can be used to recommend many types of items, such as web pages (URLs), news, videos, images, etc. The system maintains information about user preferences either by initial input about users' interests during the registration process or by rating documents. Recommendations are then formed by taking into account the content of documents and by filtering in the ones that better match the users' preferences and logged profile. For example, in a movie recommendation application, in order to recommend movies to a user, the content-based recommendation system tries to understand the commonalities among the movies that user has rated highly in the past (specific actors, directors, genres, subject matter, etc.). Then, only the movies that have a high degree of similarity to whatever users' preferences are would be recommended.

Besides the traditional heuristics that are based mostly on information retrieval methods, other techniques for content-based recommendation have also been used, such as Bayesian classifiers [164, 143] and various machine learning techniques, including clustering, decision trees, and artificial neural networks [165]. These techniques differ from information retrieval-based approaches in that they calculate utility predictions based not on a heuristic formula, such as a cosine similarity measure, but rather are based on a model learned from the underlying data using statistical learning and machine learning techniques.

Motivated by the observation that social tagging activity not only can represent users' judgments on the resources, but also can indicate users' personal interests, we develop a novel content-based filtering framework, referred to as UserRec, to recommend users with similar interests in Chapter 4.

2.1.2 Collaborative Filtering

In this section, we review several major approaches for collaborative filtering. Two types of collaborative filtering approaches are widely studied: memory-based and model-based [30].

Memory-based Collaborative Filtering

Memory-based (or neighborhood-based) collaborative filtering approaches are widely investigated [30, 51, 63, 75, 89, 98, 115, 151, 181, 200] and employed in industrial collaborative filtering systems [115, 181, 49].

The most studied memory-based approaches include user-based approaches [30, 75, 89, 243] and item-based approaches [200, 53, 115].

The user-item matrix is usually constructed in collaborative filtering. Suppose a recommender system has *M* users and *N* items, the relationship between users and items is denoted by a $M \times N$ matrix, referred to as the user-item matrix. Each entry in this matrix $r_{m,n}$ represents the rating that user m rates item n, where $r_{m,n} \in \{1, 2, \ldots, r_{max}\}$. If the user m does not rate the item n, $r_{m,n} = 0$.

Pearson correlation coefficient (PCC) [181] and vector space model (VSS) [30] are often applied in memory-based algorithms. PCC-based collaborative filtering generally can achieve higher performance than the other popular algorithm VSS, since it considers the differences of user rating styles [127].

User-based methods look for some similar users who have similar rating styles with the active user and then employ the ratings from those similar users to predict the ratings for the active user [206]. The original PCC equation is as follows:

$$Sim(a, u) = \frac{\sum\limits_{i=1}^{n} (r_{a,i} - \bar{r}_a) \cdot (r_{u,i} - \bar{r}_u)}{\sqrt{\sum\limits_{i=1}^{n} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum\limits_{i=1}^{n} (r_{u,i} - \bar{r}_u)^2}}. \tag{2.1}$$

In user-based collaborative filtering, PCC is usually employed to define the similarity between two users a and u based on the items they rated in common:

$$Sim(a,u) = \frac{\sum\limits_{i \in I(a) \bigcap I(u)} (r_{a,i} - \bar{r}_a) \cdot (r_{u,i} - \bar{r}_u)}{\sqrt{\sum\limits_{i \in I(a) \bigcap I(u)} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum\limits_{i \in I(a) \bigcap I(u)} (r_{u,i} - \bar{r}_u)^2}}, \tag{2.2}$$

where $Sim(a,u)$ is the similarity between user a and user u, and i belongs to the subset of items which user a and user u co-rated. $I(a)$ is the item set rated by user a, and $I(u)$ is the item set rated by user u. $r_{a,i}$ is the score user a gave to item i, and $\bar{r}_a$ represents the average score of user a. $Sim(a,u) \in [-1,1]$, a larger value means user a and u are more similar, and a smaller value means user a and u are less similar. In the VSS approach, user u and user a are considered as vectors, and cosine similarity is employed to compute similarity.

$$\begin{aligned} Sim(a,u) &= cos(\vec{a},\vec{u}) = \frac{\vec{a} \cdot \vec{u}}{||\vec{a}||_2 \times ||\vec{u}||_2} \qquad (2.3) \\ &= \frac{\sum\limits_{i \in I(a) \bigcap I(u)} r_{a,i} \cdot r_{u,i}}{\sqrt{\sum\limits_{i \in I(a) \bigcap I(u)} r_{a,i}^2} \times \sqrt{\sum\limits_{i \in I(a) \bigcap I(u)} r_{u,i}^2}}, \end{aligned}$$

where $Sim(a,u)$ is the similarity between user a and user u, $cos(\vec{a},\vec{u})$ is the cosine similarity, $\vec{a}$ is the vector representation of user a, $\vec{a} \cdot \vec{u}$ is the dot product between the vectors $\vec{a}$ and $\vec{u}$.

After calculating similarities between two users, we can predict the value of missing rating $r_{u,i}$ of user u to item i by considering k most similar users for item i. Adomavicius and Tuzhilin [2] presented several functions for aggregating ratings of the k most similar users:

$$r_{u,i} = \frac{1}{k} \sum_{u' \in U'} r_{u',i}, \tag{2.4}$$

$$r_{u,i} = \frac{1}{\sum\limits_{u' \in U'} |Sim(u,u')|} \sum_{u' \in U'} Sim(u,u') \times r_{u',i}, \tag{2.5}$$

$$r_{u,i} = \bar{r}_u + \frac{1}{\sum\limits_{u' \in U'} |Sim(u,u')|} \sum_{u' \in U'} Sim(u,u') \times (r_{u',i} - \bar{r}_{u'}), \quad (2.6)$$

where U' denotes the set of k users who are the most similar to the user u and who rated item i.

Item-based approaches share similar idea with user-based methods except for predicting the ratings of active users based on the information of similar items computed [53, 200]. Given two item i and j, to compute PCC in item-based approaches, we need to find users who rated both items in the past. The equation of employing PCC to compute similarity between item i and item j is as follows:

$$Sim(i,j) = \frac{\sum\limits_{u \in U(i) \bigcap U(j)} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sqrt{\sum\limits_{u \in U(i) \bigcap U(j)} (r_{u,i} - \bar{r}_i)^2} \cdot \sqrt{\sum\limits_{u \in U(i) \bigcap U(j)} (r_{u,j} - \bar{r}_j)^2}}, \quad (2.7)$$

where $Sim(i,j)$ is the similarity score between item i and item j, $U(i)$ is the user set who rated by item i, $U(j)$ is the user set who rated item j, $r_{u,i}$ is score user u rated item i, $r_{u,j}$ is score user u rated item j, $\bar{r}_i$ is the average score of item i. $Sim(i,j) \in [-1,1]$, a larger value means two items i and j are more similar, and a smaller value means two items i and j are less similar. After calculating similarity between two items i and j, it is similar to user-based approach to compute missing rating $r_{u,i}$ of user u to item i.

Model-based Collaborative Filtering

Different from memory-based collaborative filtering, model-based approaches first train a model based on observed user-item ratings, and then employ the trained model to predict missing values [26, 35, 61, 64, 137, 162, 231]. Billsus et al. [26] proposed to use machine learning techniques with feature extraction to tackle collaborative filtering. Canny et al. [35] aimed at protecting privacy of individual data through leveraging a peer-to-peer protocol. Canny et al. [35] also proposed a factor analysis approach that has advantages in speed and storage over previous algorithms.

Aspect models [80, 208], Bayesian model [42], relevance models [230, 232], latent class models [81, 91, 138, 208, 201], matrix factorization models [26, 64, 180, 199] and clustering models [12, 60, 97, 163, 226, 225] also belong to the model-based collaborative filtering. Wang et al. [232] proposed a probabilistic user-to-item relevance framework that introduces the concept of relevance into the related problem of collaborative filtering. Experimental results complement the theoretical insights with improved recommendation accuracy. Different types of ratings are used so that the unified model is more robust to data sparsity. Jin et al. [91] conducted a broad and systematic study on different mixture models for collaborative filtering. That work discussed general issues related to using a mixture model for collaborative filtering, and proposed three properties that a graphical model is expected to satisfy. Using those properties, they thoroughly examined five different mixture models, including Bayesian Clustering (BC), Aspect Model (AM), Flexible Mixture Model (FMM), Joint Mixture Model (JMM), and the Decoupled Model (DM). They compared those models both analytically and experimentally. Experiments over two data sets of movie ratings under different configurations show that in general, whether a model satisfies the proposed properties tends to be correlated with its performance. In particular, the Decoupled Model, which satisfies all the three desired properties, outperforms the other mixture models as well as many other existing approaches for collaborative filtering. Their study showed that graphical models are powerful tools for modeling collaborative filtering, but careful design is necessary to achieve good performance. Kohrs et al. [97] presented an algorithm for collaborative filtering based on hierarchical clustering, which tried to balance both robustness and accuracy of predictions, especially when few data were available. Shani et al. [204] used Markov decision processes for generating recommendations since they treated the recommendation process as a sequential decision problem. Marlin et al. [137] proposed to use Latent Dirichlet Allocation through a combination method of multinomial mixture and aspect model. Si et al. [208] used probabilistic latent semantic model to propose a flexible mixture model that allows modeling the classes of users and items explicitly with two sets of latent variables. Kumar et al. [104] used a simple probabilistic model to demonstrate that collaborative filtering is valuable with relatively little data on each user, and in certain restricted settings, simple collaborative filtering algorithms are almost as effective as the best possible algorithms in terms of utility.

Recently, low-dimensional matrix factorization approaches have been studied since their efficiency in dealing with large scale data sets. Matrix factorization methods first learn a compact model based on observed data, and then apply it to predict missing values [180, 192, 193, 218, 130, 132, 127].

Singular value decomposition (SVD) is the basic approach to solve low-rank matrix factorization through minimizing the sum-of-the-squared error [65]. A simple and efficient expectation maximization (EM) algorithm for solving weighted low-rank approximation is proposed in [218]. Srebro et al. [219]proposed a matrix factorization approach to constrain the norms of U and V instead of their dimensionality. Salakhutdinov et al. [193] proposed probabilistic matrix factorization with Gaussian noise in each observation. In [192], the Gaussian-Wishart priors are placed on the user and item hyperparameters. Low-dimensional methods are shown to be effective and efficient, but these methods suffer several shortcomings. SVD method [65], as well as other well-known methods such as weighted low-rank approximation [218], probabilistic principal component analysis (PPCA) [224], probabilistic matrix factorization (PMF), and constrained probabilistic matrix factorization [193], the latent features are hard to interpret, and there is no range constraint bound on the latent feature vectors. Zhang et al. [253] proposed nonnegative matrix factorization (NMF) by imposing nonnegative constraints on user-specific features U and item-specific features V. Pan et al. [159] employed transfer learning in collaborative filtering with uncertain ratings by integrative factorization.

Learning two compact latent feature spaces of user space U and item space V is a fundamental problem to low-rank matrix factorization. Regularized matrix factorization is usually employed. Given an $m \times n$ user-item rating matrix R, the low-rank matrix factorization method tries to fit it using $R = U^T V$, where $U \in \mathcal{R}^{l \times m}$, $V \in \mathcal{R}^{l \times n}$, and U^T is the transpose of matrix U. We need to learn U and V by learning with observation data. The physical meaning of each latent factor is a preference vector. A user's preference is represented with a linear combination of each preference vector, with user-specific co-efficient. An item's property corresponds to a linear combination of each preference vector, with item-specific co-efficient. We can solve the minimization problem to get U and V:

$$\begin{aligned} \min_{U,V} \mathcal{L}(R, U, V) &= \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij}^R (R_{ij} - U_i^T V_j)^2 \\ &+ \frac{\lambda_U}{2} ||U||_F^2 + \frac{\lambda_V}{2} ||V||_F^2, \end{aligned} \tag{2.8}$$

where $\mathcal{L}(R, U, V)$ is the loss function, I_{ij}^R is the indicator function that equals to 1 if user u_i rated item v_j and equals to 0 otherwise, $|| \cdot ||_F^2$ is the Frobenius norm. A local minimal of the objective function could be found by performing gradient descent on U_i and V_j:

$$\frac{\partial \mathcal{L}}{\partial U_i} = \sum_{j=1}^{n} I_{ij}^R (U_i^T V_j - R_{ij}) V_j + \lambda_U U_i, \tag{2.9}$$

$$\frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{m} I_{ij}^R (U_i^T V_j - R_{ij}) U_i + \lambda_V V_j. \tag{2.10}$$

To tackle the data sparsity problem and non-flexibility problem confronted by traditional collaborative filtering algorithms, we develop a unified matrix factorization framework through fusing users' rating information and tagging information in Chapter 3.

2.2 Information Retrieval Models

In information retrieval, a key research challenge is to seek an optimal ranking function, which is usually based on a retrieval model. The retrieval model formally defines the notion of relevance and enables us to derive a retrieval function that can be computed to score and rank documents.

Classic information retrieval models include Boolean model [223, 148, 108, 82], vector space model [194, 272], probabilistic model [207, 139, 187, 57, 45, 216, 185, 189], language model [170, 77, 144, 47, 170, 249, 248, 252, 101, 118, 222, 106] and translation model [21, 31, 88, 154, 184]. The boolean model [67, 197] is a simple retrieval model based on set theory and Boolean algebra, in which documents and queries are represented as sets of index terms. Boolean model does not have the score for each query-document pair, thus not suitable in Web search or social media systems [11]. We mainly introduce vector space model [93, 198, 240, 196], probabilistic model and language model [106, 188, 216, 107, 170, 251, 252, 249], and translation model in this thesis [31, 88, 154].

2.2.1 Vector Space Model

In the vector space model [194, 198, 240], documents and queries are represented as vectors in a common M-dimensional space. Each dimension corresponds to a separate term. The definition of term is application-dependent, usually it could be original single word, stemmed single word, n-gram, keyword or phrase. Typically the unigram words are considered as terms, and the dimensionality M of the vector is the number of words in the vocabulary of the application. If a term t appears in a document d_j, the term is usually associated with a non-zero weight w_{t,d_j} in the document vector space $\vec{d_j}$. These term weights are then employed to compute the similarity between a user query and a document. Luhn [125, 126] described some of the earliest reported applications of term weighting.

There are many weights proposed to compute these term weights, and one of the best well known approach is term frequency-inverse document frequency (TF-IDF) weighting [93, 196]. Term frequency (TF) of a term t in a document d_j, tf_{t,d_j} is denoted as the number of occurrences of term t in the document d_j. For a document d_j, the set of weights determined by the tf_{t,d_j} weights above may be viewed as a quantitative digest of that document. In this view of a document, known in the literature as the bag-of-words model, the exact ordering of the terms in a document is ignored but the number of occurrences of each term is material. We only retain information on the number of occurrences of each term. Raw term frequency as above suffers from a critical problem: all terms are considered equally important when it comes to assessing relevance on a query. In fact, certain terms have little or no discriminating power in determining relevance. For instance, a collection of documents on the *machine learning* topic is likely to have the term *learning* in almost every document. Thus, inverse document frequency (IDF) is introduced as a mechanism for attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination.

Document frequency (DF) of a term t in a collection, df_t is defined to be the number of documents in the collection that contain the term t. Denoting the total number of documents in a collection by N, the inverse document frequency (IDF) of a term t is defined as follows:

$$idf_t = \log \frac{N}{df_t}. \tag{2.11}$$

Sparck Jones [93] showed the use of inverse document frequency in term weighting through detailed experiments. A series of extensions and theoretical justifications of IDF are due to Salton and Buckley [196], Robertson and Jones [187], Croft and Harper [46], and Papineni [161].

TF-IDF is to combine the definitions of term frequency and inverse document frequency to produce a composite weight for each term in each document. The TF-IDF weighting scheme assigns to term t a weight in document d_j as follows:

$$tf - idf_{t,d_j} = tf_{t,d_j} \times idf_t. \tag{2.12}$$

Thus, TF-IDF is higher when t occurs many times within a small number of documents, lower when the term occurs fewer times in a document or occurs in many documents, and lowest when the term occurs in virtually all documents.

There are variants of TF-IDF functions. Two famous ones are sub-linear TF scaling and maximum TF normalization. It seems unlikely that thirty occurrences of a term in a document truly carry thirty times the significance of a single occurrence. A sub-linear TF scaling is to use logarithm instead of the raw term frequency:

$$wf_{t,d_j} = \begin{cases} 1 + \log tf_{t,d_j} & \text{if } tf_{t,d_j} > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.13}$$

where wf_{t,d_j} is the sub-linear TF scaling.

Maximum TF normalization is a well-studied technique to normalize the TF weights of all terms occurring in a document by the maximum tf_{t,d_j} in the document. For each document d, let $tf_{max}(d_j) = \max_{t' \in d_j} tf_{t',d_j}$, where t' ranges over all terms in d_j. Then we compute a normalized term frequency for each term t as follows:

$$ntf_{t,d_j} = a + (1 - a)\frac{tf_{t,d_j}}{tf_{max}(d_j)}, \tag{2.14}$$

where a is a value between 0 and 1 and is generally set to 0.4, and it is a smoothing term whose role is to damp the contribution of the second term.

The document vector $\vec{d_j}$ is defined as $\vec{d_j} = (w_{1,d_j}, w_{2,d_j}, \ldots, w_{M,d_j})$, and the query vector $\vec{q_i} = (w_{1,q_i}, w_{2,q_i}, \ldots, w_{M,q_i})$. w_{k,d_j} and w_{k,q_i} are usually TF-IDF values or variants of TF-IDF. The vector space model evaluates the similarity of a user query q_i and a document d_j based on vectors of $\vec{q_i}$ and $\vec{d_j}$. Cosine similarity is usually employed:

$$\begin{aligned} Sim(q_i, d_j) &= \frac{\vec{q_i} \cdot \vec{d_j}}{|\vec{q_i}| \times |\vec{d_j}|} \\ &= \frac{\sum\limits_{k=1}^{M} w_{k,q_i} \times w_{k,d_j}}{\sqrt{\sum\limits_{k=1}^{M} w_{k,q_i}^2} \times \sqrt{\sum\limits_{k=1}^{M} w_{k,d_j}^2}}, \qquad (2.15) \end{aligned}$$

where $|\vec{q_i}|$ and $|\vec{d_j}|$ are the norms of query and document respectively. The basic computation of cosine scores is due to Zobel and Moffat [272].

Vector space model with TF-IDF weighting and document length normalization [212] has proven to be one of the most effective retrieval models [136].

2.2.2 Probabilistic Model and Language Model

In the probabilistic model, the process of document retrieval could be treated as estimating the probability that a document is relevant to a query [106, 188, 217]. The *probability ranking principle* (PRP) is an important concept for probabilistic model [186]. Under a ranked retrieval setup assumption, where there is a collection of documents, the user issues a query, and an ordered list of documents is returned. A binary notion of relevance assumption is also needed. For a query q and a document d in the collection, let $R_{d,q}$ be an indicator random variable that says whether d is relevant with respect to a given query q. It takes on a value of 1 when the document is relevant and 0 otherwise. In this section, we write R instead of $R_{d,q}$ for short.

Using a probabilistic model, the obvious order in which to present documents to the user is to rank documents by their estimated probability of relevance with respect to the information need:$P(R = 1|d, q)$. This is the basis of PRP [45].

The *binary independence model* (BIM) is the model that has traditionally been used with the PRP [187, 57]. Documents and queries are both represented as binary term incidence vectors. A document d is represented by the vector $\vec{x} = (x_1, x_2, \ldots, x_M)$ where $x_t = 1$ if term t is present in document d and $x_t = 0$ if t is not present in d. Similarly, q is represented by the incidence vector $\vec{q}$. It is assumed that the relevance of each document is independent of the relevance of other documents. Under the BIM, the probability $P(R|d, q)$ that a document is relevant is modeled via the probability in terms of term incidence vectors $P(R|\vec{x}, \vec{q})$. Using Bayes rule, the equations are:

$$P(R = 1|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 1, \vec{q})P(R = 1|\vec{q})}{P(\vec{x}|\vec{q})}, \tag{2.16}$$

$$P(R = 0|\vec{x}, \vec{q}) = \frac{P(\vec{x}|R = 0, \vec{q})P(R = 0|\vec{q})}{P(\vec{x}|\vec{q})}, \tag{2.17}$$

where $P(\vec{x}|R = 1, \vec{q})$ and $P(\vec{x}|R = 0, \vec{q})$ are the probability that if a relevant or non-relevant document is retrieved, then that document's representation is $\vec{x}$. $P(R = 1|\vec{q})$ and $P(R = 0|\vec{q})$ indicate the prior probability of retrieving a relevant or non-relevant document for a query q. Then, documents ar ranked by their odds of relevance:

$$\begin{aligned} O(R|\vec{x}, \vec{q}) &= \frac{P(R = 1|\vec{x}, \vec{q})}{P(R = 0|\vec{x}, \vec{q})} = \frac{\frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})}}{\frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}} \\ &= \frac{P(R = 1|\vec{q})}{P(R = 0|\vec{q})} \cdot \frac{P(\vec{x}|R = 1, \vec{q})}{P(\vec{x}|R = 0, \vec{q})}, \end{aligned} \tag{2.18}$$

Naive Bayes conditional independence assumption is made that the presence or absence of a word in a document is independent of the presence or absence of any other word:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}, \tag{2.19}$$

because x_t is either 0 or 1, the equation could be written as:

$$O(R|\vec{x},\vec{q}) = O(R|\vec{q}) \cdot \prod_{t:x_t=1} \frac{P(x_t=1|R=1,\vec{q})}{P(x_t=1|R=0,\vec{q})} \cdot \prod_{t:x_t=0} \frac{P(x_t=0|R=1,\vec{q})}{P(x_t=0|R=0,\vec{q})}. \tag{2.20}$$

Let $p_t = P(x_t = 1|R = 1, \vec{q})$ be the probability of a term appearing in a document relevant to the query, and $u_t = P(x_t = 1|R = 0, \vec{q})$ be the probability of a term appearing in a non-relevant document. Make an additional assumption that terms no occurring in the query are equally likely to occur in relevant and non-relevant documents, then it is only needed to consider terms in the products that appear in the query:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}, \tag{2.21}$$

Manipulate this expression by including the query terms found in the document into the right product, but simultaneously dividing through by them in the left product:

$$O(R|\vec{q},\vec{x}) = O(R|\vec{q}) \cdot \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} \cdot \prod_{t:q_t=1} \frac{1-p_t}{1-u_t}, \tag{2.22}$$

the right product is over all query terms, thus is a constant for a query like the odds $O(R|\vec{q})$. Then the only quantity that needs to be estimated to rank documents for relevance to a query is the left product. Taking the logarithm of this term, the resulting quantity used for ranking is called the *retrieval status value* (RSV):

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}. \tag{2.23}$$

Over the decades, an interesting class of probabilistic models called language modeling approaches have been developed. The language modeling approach was first proposed by Ponte and Croft in [108]. The goal is to infer a language model for each document and rank documents according to the probability $P(q|d)$ of the given query q. The original and basic method for using language models in IR is the query likelihood model [249]. A language model M_d is constructed from each document d in the collection. The goal is to rank documents by $P(d|q)$, where the probability of a document is interpreted as the likelihood that it is relevant to the query. Using Bayes rule, the equation is as follows:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)}, \tag{2.24}$$

where $P(q)$ is the same for all documents, and can be ignored. The prior probability of a document $P(d)$ is often treated as a uniform across all d in collection and it can also be ignore [256, 255]. Thus, results are ranked by $P(q|d)$, the probability of the query q under the language model derived from d. The language modeling approaches attempts to model the query generation process: documents are ranked by the probability that a query would be observed as a random sample from the respective document model. The probability of generating the query given the language model M_d of document d using maximum likelihood estimation (MLE) and the unigram assumption is:

$$\hat{P}(q|M_d) = \prod_{t \in q} \hat{P}_{mle}(t|M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d}, \tag{2.25}$$

where M_d is the language model of document d, $tf_{t,d}$ is the raw term frequency of term t in document d, and L_d is the number of tokens in document d.

Many variants of the basic language modeling have been proposed and investigated, including relevance-based language model [107], title language model [90], cluster-based language model [118], etc. To avoid the zero probability problem, a common step for language models is to perform smoothing for the unseen query terms in the document [251, 249, 252, 250]. Smoothing methods such as Jelinek-Mercer smoothing and Bayesian smoothing using Dirichlet prior have been employed [249]. Jelinek-Mercer smoothing uses a mixture between a document-specific multinomial distribution and a multinomial distribution estimated from the entire collection:

$$\hat{P}(t|d) = \lambda \hat{P}_{mle}(t|M_d) + (1-\lambda)\hat{P}_{mle}(t|M_c), \tag{2.26}$$

where $0 < \lambda < 1$ and M_c is a language model built from the entire document collection.

Another popular smoothing method is Dirichlet smoothing, and the equation is as follows:

$$\hat{P}(t|d) = \frac{tf_{t,d} + \alpha \hat{P}(t|M_c)}{L_d + \alpha}, \tag{2.27}$$

where α is a parameter, and typically is set to 2000.

2.2.3 Translation Model

Translation model, originated from machine translation, has been employed in ad-hoc retrieval, FAQ retrieval and question search. Riezler et al. [184] proposed a translation model for question search in FAQ. Their translation model was trained on a large amount of data extracted from FAQ pages on the Web. Jeon et al. [87, 86] employed translation model to tackle the question search problem in community-based Q&A. Translation model, proposed by Berger et al. [21], has been extensively employed in question finding and answer retrieval [20, 54, 88, 184]. Realizing that translation model may produce inconsistent probability estimates and make the model unstable, Xue et al. [244] proposed translation-based language model which balanced between language model and translation model. Learning monolingual word to word translation probability is the most essential part of solving the lexical gap problem in translation-based models. It can be obtained by training statistical translation models on parallel monolingual corpora. IBM model 1 was commonly employed to learn the translation probabilities [31]. Jeon, Croft and Lee considered question-question pairs as a parallel corpus if their answers are similar [87, 86]. Xue, Jeon and Croft treated question-answer pairs as a parallel corpus [244]. Bernhard and Gurevych proposed to use a parallel training dataset of the definitions and glosses provided for the same term by different lexical semantic resources [23].

To measure questions' semantic relatedness and solve the lexical gap problem encountered by bag-of-words approaches, we propose two models to combine lexical information with latent semantic knowledge in Chapter 5.

2.3 Machine Learning

Machine learning techniques are employed in this thesis to help recommend in social media. In this section, we will review three most representative machine learning techniques, namely supervised learning, semi-supervised learning and unsupervised learning.

2.3.1 Supervised Learning

In supervised learning, we are given a set of N independent and identically distributed (i.i.d) data sampled from a fixed but unknown distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$ where:

$$D = \{(x_i, y_i)\}_{i=1}^{N}, x_i \in \mathcal{X}, y_i \in \mathcal{Y}, \tag{2.28}$$

where $\mathcal{X} \in \mathcal{R}^d$ is the input space, and $\mathcal{Y}$ is the output space. $\mathcal{Y}$ is a small number of discrete classes for classification problem and $\mathcal{Y} \in \mathcal{R}$ for regression problem.

The objective of supervised learning is to seek a function $f : \mathcal{X} \mapsto \mathcal{Y}$ that maps inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$ while $f(x)$ approximates y on new samples from the distribution $(x, y) \sim \mathcal{P}$. The detailed steps of supervised learning procedure are as follows:

1. Propose hypothesis $\mathcal{H}$: $f : \mathcal{X} \mapsto \mathcal{Y}$
2. Collect labeled training data set: $\{x_i, y_i\}_{i=1}^{N}$
3. Train a supervised learning model: find $f \in \mathcal{H}$ that satisfies $f(x_i) \approx y_i$
4. Predict on new data sample: $\hat{y}_i = f(x_i)$

The problem of supervised learning has a long history and has attracted a lot of contributions [55, 72, 146, 227]. The two most important factors of variation of supervised learning algorithms are the hypothesis class $\mathcal{H}$ and the criterion for selecting f from $\mathcal{H}$ given the training data. A standard criterion is to quantify what it means for $f(x)$ to approximate y, which is measured by the expected error of approximation by the risk functional $\mathcal{R}^l_{\mathcal{P}}[f]$ defined as:

$$\mathcal{R}^l_{\mathcal{P}}[f] = E_{(x,y)\sim\mathcal{P}}[l(x, y, f(x))], \tag{2.29}$$

where the loss function $l : (\mathcal{X}, \mathcal{Y}, \mathcal{Y}) \mapsto \mathcal{R}$ measures the penalty for predicting $f(x)$ on the sample (x, y). In general, $l(x, y, \hat{y}) = 0$ if $y = \hat{y}$. Many loss functions are available, such as $0/1$ loss [55], logit loss [83], hinge loss [227], etc.

Since the distribution $\mathcal{P}$ is generally unknown, one estimates the risk of f using its empirical risk $\mathcal{R}^l_{\mathcal{D}}$, computed on the training set D as follows:

$$\mathcal{R}^l_{\mathcal{D}}[f] = \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i, f(x_i)), \tag{2.30}$$

where $\mathcal{R}^l_{\mathcal{D}}[f]$ is training error.

Minimizing empirical risk on the training data may bring the problem of overfitting. Thus, a validation set is usually needed to complement the training set to find the best hypothesis.

Due to the importance of supervised learning, many models have been proposed, such as Naive Bayes [142], decision tree [175], support vector machines [73], etc.

Feature selection is an important step in supervised learning. Zhu et al. [270] studied automated graph classification problem, identified two main issues with the most widely used feature selection approach which is based on a discriminative score to select frequent subgraph features, and introduced a new diversified discriminative score to select features that have a higher diversity.

2.3.2 Semi-Supervised Learning

Semi-supervised learning [40, 266] tries to learn from both labeled data and unlabeled data. Recently, many models have been proposed to achieve semi-supervised learning, including co-training [28], self-training [191], transductive support vector machines [92, 211], and graph-based approaches [14, 15, 214, 258, 268, 269].

A co-training approach identifies a problem setting from different views, trains a model on each view, and enlarges the training set of other views based on each view. For example, in the task of learning to classify Web pages, in which the description of each example can be partitioned into two distinct news. The description of a Web page can be partitioned into the words occurring on that page, and the words occurring in hyperlinks that point to that page. It is assumed that either view of the example would be sufficient for learning if there are enough labeled data. But the goal of co-training is to use both views together to allow inexpensive unlabeled data to augment a much smaller set of labeled examples. Specifically, the presence of two distinct views of each example suggests strategies in which two learning algorithms are trained separately on each view, and then each algorithm's predictions on new unlabeled examples are used to enlarge the training of the other [28].

In self-training [191], or incremental training [153], an initial model is constructed by using the fully labeled data. This model is used to estimate labels for the weakly labeled or unlabeled data. A selection metric is then used to decide which of the weakly labeled examples were labeled corrected. Those examples are then added to the training set and the process repeats. Regular support vector machines (SVMs) try to induce a general decision function for a learning task. Transdutive support vector machines take into account a particular test set and try to minimize misclassifications of just those particular examples [92, 211].

The graph-based semi-supervised learning can be modeled as a random walk with label propagation from labeled data to unlabeled data [267, 268]. From a different perspective, this method could be viewed as having a quadratic loss function with infinity weight, so that the labeled data are fixed at given label values, and a regularizer based on the graph information:

$$R = \frac{1}{2}\sum_{i,j}^{n} w_{ij}(f_i - f_j)^2 + \sum_{i \in L}(f_i - y_i)^2, \qquad (2.31)$$

where w_{ij} corresponds to the weight between point i and point j, L is the set of labeled data, and y_i is the label value. In the equation, the second component only considers the loss function using the labeled data. The local and global consistency method proposed by Zhou et al. [258] used the loss function based on both labeled and unlabeled data, and the normalized graph Laplacian in the regularizer:

$$R = \frac{1}{2}\sum_{i,j}^{n} w_{ij}(\frac{f_i}{\sqrt{D_{ii}}} - \mu\frac{f_i}{\sqrt{D_{jj}}})^2 + \sum_{i}^{n}(f_i - y_i)^2, \qquad (2.32)$$

where D is a diagonal matrix with entries $D_{ii} = \sum\limits_{j}(w_{ij})$, and $\mu > 0$ is the regularization parameter. The first term of the right-hand side in the cost function is the smoothness constraint, which means that a good classifying function should not change too much between nearby points. The second term is the fitting constraint, which means a good classifying function should not change too much from the initial label assignment. The trade-off between these two competing constraints is captured by the parameter μ. By minimizing the cost function R, the solution can be obtained which is equivalent to that of the iterative label propagation algorithm.

2.3.3 Unsupervised Learning

Unsupervised learning means that there is no human expert who has assigned documents to classes. Clustering is the most common form of unsupervised learning. In text domain, clustering algorithms group a set of documents into subsets or clusters, and the algorithm's goal is to create clusters that are coherent internally, but clearly different from each other.

Berkhin [22] gave a general up-to-date survey of clustering methods with special attention to scalability. The classic reference for clustering in pattern recognition, covering both K-means and EM, is [55]. Rasmussen [71] introduced clustering from an information retrieval perspective. Anderberg [5] provided a general introduction to clustering for applications. In addition to Euclidean distance and cosine similarity, Kullback-Leibler divergence is often used in clustering as a measure of how similar documents and clusters are [242, 149, 105]. Cheng et al. [41] proposed an efficient incremental computing approach to cluster large attributed information networks.

Traditional clustering algorithms mainly discuss hard clustering, in which each object only belongs to a cluster or not. Recently, soft clustering, e.g., latent topic modeling, in which each object belongs to each cluster to a certain degree has become popular. Latent Dirichlet Allocation (LDA) [27], which is used to build topic models based on a formal generative model of documents [74] is heavily cited in the machine learning literature due to its property of possessing fully generated semantics. Wei et al. [235] proposed a LDA-based document model within language modeling framework for ad-hoc retrieval.

We develop a supervised approach to solve the question subjectivity identification problem in chapter 6. Specifically, we propose an approach to collect training data automatically by utilizing social signals without involving any manual labeling.

2.4 Rating Prediction

Rating prediction is an important application in social media systems. We apply proposed models in rating prediction in chapter 3. In social media systems, users could rate items to express their preferences, items could be movie, book, product, restaurant, etc. Examples of social media systems that employ rating prediction include Amazon[1], MovieLens[2], Netflix[3], etc. A formal description of the problem of rating prediction is as follows: there are m users and n items. Matrix $R \in \mathcal{R}^{m \times n}$ is the rating matrix, where r_{ij} is the score that user u_i rated v_j, R is a partially filled matrix. The target of rating prediction is to fill the rating matrix R based on existing observed ratings in R, so that given an active user u_i, we can predict the rating on an item $v_{j'}$ that the user has not rated before.

The Netflix Prize competition is an important event related to rating prediction[4]. It is started and supported by Netflix, a company providing online movie rental services in USA. In October 2006, Netflix released a large movie rating dataset containing about 100 million ratings from over 480 thousand randomly selected customers on nearly 18 thousand movie items. Root mean square error (RMSE)

[1] http://www.amazon.com

[2] http://movielens.umn.edu

[3] http://www.netflix.com

[4] http://en.wikipedia.org/wiki/Netflix_Prize

is employed for performance evaluation:

$$RMSE = \sqrt{\frac{\sum_{ij}(r_{ij} - \hat{r}_{ij})}{N}}, \tag{2.33}$$

where r_{ij} is the real rating of user u_i to item i_j, $\hat{r}_{ij}$ is the predicted rating by some model, N denotes the number of tested ratings. A lot of concepts and approaches have been proposed during Netflix competition [17, 18, 19, 99, 100, 192, 193, 124, 247, 265].

Two metrics are commonly used evaluating rating prediction, one is RMSE introduced before, the other is mean absolute error (MAE). The metric MAE is defined as:

$$MAE = \frac{\sum_{ij} |r_{ij} - \hat{r}_{ij}|}{N}, \tag{2.34}$$

where r_{ij} is the real rating of user u_i to item i_j, $\hat{r}_{ij}$ is the predicted rating by some model, N denotes the number of tested ratings.

Besides MAE and RMSE, Liu et al. [117] proposed a collaborative filtering approach that addresses the item ranking problem directly by modeling user preferences derived from the ratings. They measured the similarity between users based on the correlation between their rankings of the items rather than the rating values and proposed new collaborative filtering algorithms for ranking items based on the preferences of similar users. Experimental results on real world movie rating data sets showed that the proposed approach outperformed traditional collaborative filtering algorithms significantly on the NDCG measure for evaluating ranked results.

Traditional collaborative filtering approaches such as memory-based and model-based methods surveyed above could be applied on rating prediction. With the development of social media systems, huge amount of user-generated content (UGC) in social media systems has brought rating prediction to a new era.

Singla and Richardson [213] analyzed the who talks to whom social network on the MSN instant messenger over 10 million people with their related search records on the Live Search Engine, and revealed that people who chat with each other are more likely to share interests. Based on this finding, many researchers have started to investigate trust-based rating prediction on social media systems [6, 13, 29, 62, 85, 140, 141, 157, 155, 156, 160, 168, 167, 236, 229]. [140] proposed a trust-aware collaborative filtering method for recommender systems. The collaborative filtering process is informed by the reputation of users which is computed by propagating trust. Trust values are computed in addition to similarity measures between users. Bedi et al. [13] proposed a trust-based recommender system for the Semantic Web. Ma et al. [127] proposed to recommend with social trust ensemble. Ma et al. [128] proposed to recommend with explicit and implicit social relations. Ma et al. [129] proposed to recommend with social regularization.

Rating prediction also needs to deal with the cold start problem as new users and/or items are always present. Rating elicitation is a common approach for handling cold start. Liu et al. [116] proposed a principled approach to identify representative users and items using representative-based matrix factorization.

2.5 User Recommendation

In social media systems, users post articles, rate movies, establish friendship links, write reviews, tag items, etc. All these content in social media systems from users is referred to as user generate content (UGC) [103]. Previous studies found that UGC are good at characterizing users' interests about Web contents [76, 112]. Thus, user recommendation, which aims at modeling and exploiting UGC to learn users' interests, and recommending users with similar interests to an active user, has become a popular application in social media systems [9, 7, 241]. User recommendation could help users discover items they may be interested in through connecting users with similar interests. We apply proposed models in user recommendation in chapter 4.

Modeling users' interests is a key challenge in user recommendation. White et al. [238] proposed to include contextual information to effectively model users' interests. Specifically, they presented a study of the effectiveness of five variant sources of contextual information for user interest model: social, historic, task, collection, and user interaction. Ma et al. [134] evaluated and compared the performance of different cues in searching and browsing activities for user interests modeling. Bila et al. [25] proposed to combine network observable data with specific queries to the user for gathering profile information about mobile phone users. Szomszor et al. [221] presented a method for the automatic consolidation of user profiles across two popular social networking sites, and subsequent semantic modeling of their interests utilizing Wikipedia as a multi-domain model. Badi et al. [10] argued that when people look through Web search results, the document triage process may involve both reading and organizing. Users' interests may be inferred from what they read and how they interact with individual documents.

After modeling users' interests, performing user similarity measure is also important. Ziegler et al. [271] analyzed correlation between trust and user similarity in online communities. Li et al. [111] proposed hierarchical-graph-based similarity measure for geographic information systems to consistently model each individual's location history and effectively measure the similarity among users. Guy et al. [68] examined nine different sources for user similarity as reflected by activity in social media applications, and suggested classification of these sources into three categories: people, things, and places.

2.6 Automatic Question Answering

2.6.1 Automatic Question Answering (Q&A) from the Web

Automatic Q&A has been a long-standing research problem which attracts contributions from the information retrieval and natural language processing communities. Automatic Q&A ranges from automatic subjective Q&A [109, 215] to automatic factual Q&A [52, 56, 70].

Most work of retrieving answers directly from the Web focus on factual Q&A. To set up a baseline for factual Q&A on the Web, how successful search engines are at retrieving accurate answers when unmodified factual natural language questions are asked was studied [178]. An architecture that augments existing search engines so that they support natural language question answering was developed [177]. To guide future system development, a specialized question answering test collection was constructed for research purpose [114]. To surmount the barrier of question understanding, the concept of a query language, which provides an intermediate system for capturing the essence of a user's information need and matching that information need to desired items in a repository of texts, was introduced [205]. In the absence of a standard query language across search engines, words were suggested to added to the question to guide the search process [3]. Determining question taxonomy is another critical component of process of machine understanding of questions. Five question taxonomies were identified at four levels of linguistic analysis [169]. Semantic enrichment of texts was studied to improve factual question answering [158].

2.6.2 Proliferation of community-based Q&A services and online forums

With the popularization of social media with Q&A aspect, people has come together to post their questions, answer other users' questions, and interact with each other. Community-based Q&A services and online forums are two representative platforms for this purpose. Overtimes, a large amount of historical Q&A pairs have been built up in their archives, providing information seekers a viable alternative to Q&A from the Web [1, 44, 84, 261].

Social cues were shown important for continuation of Q&A activity online [176]. Six classes of relevance criterion were identified for selecting best answers in community Q&A [95]. The perceived importance of relevance, quality and satisfaction in contributing to a good answer was explored [203]. An approach based on structuration theory and communities of practice that could guide investigation of dynamics of community Q&A was proposed [190]. A review and analysis of the research literature in social Q&A was conducted [59]. The motivational factors affecting the quantity and quality of voluntary knowledge contribution in community-based Q&A services was investigated [120, 122, 121].

2.6.3 Automatic Question Answering (Q&A) in social media

We apply proposed models related to automatic Q&A in chapter 5 and chapter 6. With the proliferation of community-based Q&A services and online forums, a large amount of historical Q&A pairs have been built up in social media systems. Researchers have been investigating automatic question answering in social media systems recently. Question search aims at finding semantically equivalent questions for a user question. Addressing the lexical chasm problem between user questions and the questions in a Q&A archive is the focus of most existing work. Berger et al. [20] studied four statistical techniques for bridging the lexical chasm, which include adaptive TFIDF [179], automatic query expansion [147], statistical translation models [21], and latent semantic models [79]. The history of question search originated from FAQ retrieval. The FAQ Finder combined lexical similarity and semantic similarity between questions to rank FAQs, where a vector space model was employed to compute the lexical similarity and the WordNet [145] was utilized to capture the semantic similarity [34]. Riezler et al. [184] proposed a translation model for question search in FAQ. Their translation model was trained on a large amount of data extracted from FAQ pages on the Web.

Recently, question search has been re-visited with the Q&A data in social media, which mainly includes community-based Q&A services and online forums. Jeon et al. [87, 86] employed translation model to tackle the question search problem in community-based Q&A. Translation model, proposed by Berger et al. [21], has been extensively employed in question finding and answer retrieval [20, 54, 88, 184]. Realizing that translation model may produce inconsistent probability estimates and make the model unstable, Xue et al. [244] proposed translation-based language model which balanced between language model and translation model. Learning monolingual word to word translation probability is the most essential part of solving the lexical gap problem in translation-based models. It can be obtained by training statistical translation models on parallel monolingual corpora. IBM model 1 was commonly employed to learn the translation probabilities [31]. Jeon, Croft and Lee considered question-question pairs as a parallel corpus if their answers are similar [87, 86]. Xue, Jeon and Croft treated question-answer pairs as a parallel corpus [244]. Bernhard and Gurevych proposed to use a parallel training dataset of the definitions and glosses provided for the same term by different lexical semantic resources [23].

Besides translation models, other approaches were also investigated. Bian et al. [24] proposed a learning framework for factual information retrieval within the community-based Q&A data. Particularly, they modeled the retrieval problem as one of learning ranking functions and then introduced an algorithm called GBRank for learning the ranking functions from a set of labeled data. Duan et al. summarized questions in a data structure consisting of question topic and question focus, and performed question search based on tree-cut model [54]. Question topic and question focus was also utilized to perform question clustering, which improves question search [38]. Cao et al. exploited category information of questions for improving the performance of question search. Specifically, they applied the approach to vector space model, Okapi BM25 model, language model, translation model and translation-based language model [36, 37]. A systematic evaluation of the performance of different classification methods on question topic classification was studied [174]. Wang et al. proposed a syntactic tree matching approach instead of a bag-of-word approach to find similar questions [233]. Cao et al. represented each question as question topic and question focus using tree-cutting approach, and employed minimum description length (MDL) for question recommendation [39].

□ **End of chapter.**

Chapter 3

Item Recommendation with Tagging Ensemble

3.1 Problem and Motivation

This chapter focuses on item recommendation with social tagging ensemble. Because of the exponential growth of information on the Web, users are in great need of effective recommendations in order to efficiently navigate through vast collections of items. Recommender Systems have been developed to suggest items that may interest users. Typically, recommender systems are based on Collaborative Filtering, which has been widely employed, such as in Amazon[1] and MovieLens[2]. Recently, [102] has shown that collaborative filtering outperformed humans on the average through comprehensive experiments. Two trends have rised in recommendation algorithm: one is memory-based algorithms [131, 254], and the other is model-based algorithms [193]. However, both types of algorithms suffer two weaknesses: (1) The recommendation performances deteriorate when the available ratings are very sparse. As claimed in [200], data sparsity is a common phenomenon in recommender systems, and the density of available ratings in commercial recommender systems is often less than 1%. (2) Almost all the traditional

[1] http://www.amazon.com

[2] http://movielens.umn.edu

recommendation algorithms only employ the user-item rating matrix information, but ignore other user behaviors, leading to the loss of flexibility.

Social tagging systems have recently emerged as a popular way for users to annotate, organize and share resources on the Web, such as Delicious[3], Flickr[4], and MovieLens. As a type of social media sites [39, 44], social tagging systems transform the Web into a participatory medium where users are actively creating, evaluating and distributing information. Previously, [76, 112, 202] have shown that tags can represent users' judgments about Web contents quite accurately, which are also good candidates to describe the resources.

In order to overcome the data sparsity problem and non-flexibility problem confronted by traditional recommendation algorithms mentioned above, this chapter proposes a factor analysis approach by utilizing both users' rating information and tagging information based on probabilistic matrix factorization, and we refer to this method as TagRec. The experimental results on MovieLens $10M/100K$ data set[5] show that our method performs better than the state-of-the-art approaches; in the meanwhile, our complexity analysis also implies that our approach can be scaled to very large data sets.

3.2 TagRec Framework

3.2.1 Preliminaries

To facilitate our discussions, Table 3.1 defines basic terms and notations used throughout in this chapter.

[3] http://delicious.com

[4] http://flickr.com

[5] http://grouplens.org/node/73

Table 3.1: Basic notations throughout this chapter.

Notation	**Description**
$US = \{u_i\}_{i=1}^{m}$	US is the set of users, u_i is the i-th user, m is the total number of users
$IS = \{i_j\}_{j=1}^{n}$	IS is the set of items, i_j is the j-th item, n is the total number of items
$TS = \{t_k\}_{k=1}^{o}$	TS is the set of tags, t_k is the k-th tag, o is the total number of tags
$l \in \mathbb{R}$	l is number of dimensions of latent feature space
$U \in \mathbb{R}^{l \times m}$	U is the user latent feature matrix
$V \in \mathbb{R}^{l \times n}$	V is the item latent feature matrix
$T \in \mathbb{R}^{l \times o}$	T is the tag latent feature matrix
$R = \{r_{ij}\}$, $R \in \mathbb{R}^{m \times n}$	R is the user-item rating matrix, r_{ij} is rating that user u_i gave to item i_j
$C = \{c_{ij}\}$, $C \in \mathbb{R}^{m \times o}$	C is the user-tag tagging matrix, c_{ik} is extent of user u_i's preference for tag t_k
$D = \{d_{jk}\}$, $D \in \mathbb{R}^{n \times o}$	D is the item-tag tagging matrix, d_{jk} is extent of how much tag t_k can represent the concept of item i_j
$\mathcal{N}(x\|\mu, \sigma^2)$	Probability density function of the Gaussian distribution with mean μ and variance σ^2

3.2.2 User-Item Rating Matrix Factorization

As shown in Table 3.1, we have m users and n items. The user-item rating matrix is denoted as R, and the element r_{ij} in R means the rating to item i_j given by user u_i, where values of r_{ij} are within the range $[0, 1]$. In recommender systems, ratings reflect users' judgments about the items, and most recommender systems use discrete rating values. Suppose the original rating values range from r_{min} to r_{max}, we use the function $f(x) = (x - r_{min})/(r_{max} - r_{min})$ as the mapping function to map the original rating values to values in the interval $[0, 1]$. As listed in Table 3.1, U denotes the user latent feature matrix, and V denotes the item latent feature matrix, with column vectors U_i and V_j denoting the l-dimensional user-specific and item-specific latent feature vectors respectively. We define the conditional distributions over the observed ratings in Eq. (3.1):

$$p(R|U,V,\sigma_R^2) = \prod_{i=1}^{m}\prod_{j=1}^{n}[\mathcal{N}(r_{ij}|g(U_i^T V_j),\sigma_R^2)]^{I_{ij}^R}, \tag{3.1}$$

where I_{ij}^R is an indicator variable with the value of 1 if user u_i rated item i_j, and 0 otherwise. The meaning of $U_i^T V_j$ is the rating user u_i gave to item i_j predicted by the model, and this is the typical matrix factorization approach. $g(x) = 1/1 + e^{-x}$ is the logistic function to map the value of $U_i^T V_j$ within the range of $[0, 1]$. Similar to [193], zero-mean spherical Gaussian priors are placed on the user and the item latent feature matrices, which are defined in Eq. (3.2):

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{i=1}^{m}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \\ p(V|\sigma_V^2) &= \prod_{j=1}^{n}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}). \end{aligned} \tag{3.2}$$

Through a Bayesian inference, the posterior distributions of U and V based only on the observed ratings are derived in Eq. (3.3):

$$\begin{aligned} & p(U,V|R,\sigma_V^2,\sigma_U^2,\sigma_R^2) \\ \propto\ & p(R|U,V,\sigma_R^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ =\ & \prod_{i=1}^{m}\prod_{j=1}^{n}[\mathcal{N}(r_{ij}|g(U_i^T V_j),\sigma_R^2)]^{I_{ij}^R} \\ \times\ & \prod_{i=1}^{m}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \times \prod_{j=1}^{n}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}). \end{aligned} \tag{3.3}$$

3.2.3 User-Tag Tagging Matrix Factorization

As listed in Table 3.1, we have m users and o tags. The user-tag tagging matrix is denoted as C, where the element c_{ik} in C represents the extent of user u_i's preference for tag t_k. Users' tagging activities indicate users' preference for tags, so the meaning of c_{ik} can be interpreted as whether the user u_i has used the tag t_k (a binary representation), or how strong the user u_i's preference is for the tag t_k (a real value representation). We represent c_{ik} in Eq. (3.4):

$$c_{ik} = g(f(u_i, t_k)), \tag{3.4}$$

where $g(\cdot)$ is the logistic function, and $f(u_i, t_k)$ represents the number of times user u_i uses tag t_k.

The idea of user-tag tagging matrix factorization is to derive two low-rank l-dimensional matrices U and T, representing the user latent feature matrix and the tag latent feature matrix respectively, based on the observed user-tag tagging matrix C. Denoting column vectors U_i and T_k as user-specific and tag-specific latent feature vectors respectively, we can define the conditional distributions over the observed user-tag tagging matrix in Eq. (3.5):

$$p(C|U, T, \sigma_C^2) = \prod_{i=1}^{m} \prod_{k=1}^{o} [\mathcal{N}(c_{ik}|g(U_i^T T_k), \sigma_C^2)]^{I_{ik}^C}, \tag{3.5}$$

where I_{ik}^C is an indicator variable with the value of 1 if user u_i has at least used tag t_k once, and 0 otherwise.

We also place the zero-mean spherical Gaussian priors, and through a Bayesian inference, we can derive the posterior distributions of U and T in Eq. (3.6):

$$\begin{aligned} & p(U, T|C, \sigma_U^2, \sigma_T^2, \sigma_C^2) \\ \propto\ & p(C|U, T, \sigma_C^2) p(U|\sigma_U^2) p(T|\sigma_T^2) \\ =\ & \prod_{i=1}^{m} \prod_{k=1}^{o} [\mathcal{N}(c_{ik}|g(U_i^T T_k), \sigma_C^2)]^{I_{ik}^C} \\ \times\ & \prod_{i=1}^{m} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{k=1}^{o} \mathcal{N}(T_k|0, \sigma_T^2 \mathbf{I}). \end{aligned} \tag{3.6}$$

3.2.4 Item-Tag Tagging Matrix Factorization

As denoted in Table 3.1, we have n items and o tags. The item-tag tagging matrix is denoted as D, and the element d_{jk} in D shows the extent of how much tag t_k can represent the concept of item i_j. Users annotate items with tags to express their judgments about items and distinguish one item from another. The meaning of d_{jk} can be interpreted as whether item i_j has been annotated with the tag t_k (a binary representation), or how strong tag t_k's representing ability is for item i_j (a real value representation). We represent d_{jk} in Eq. (3.7):

$$d_{jk} = g(h(i_j, t_k)), \tag{3.7}$$

where $g(\cdot)$ is the logistic function, and $h(i_j, t_k)$ is the number of times item i_j is annotated with tag t_k.

The idea of item-tag tagging matrix is to derive two low-rank l-dimensional matrices V and T, representing the item latent feature matrix and the tag latent feature matrix respectively, based on the observed item-tag tagging matrix D. Denoting column vectors V_j and T_k as item-specific and tag-specific latent feature vectors respectively, we can define the conditional distributions over the observed item-tag tagging matrix in Eq. (3.8):

$$p(D|V,T,\sigma_D^2) = \prod_{j=1}^{n}\prod_{k=1}^{o}[\mathcal{N}(d_{jk}|g(V_j^T T_k),\sigma_D^2)]^{I_{jk}^D}, \tag{3.8}$$

where I_{jk}^D is an indicator variable with the value of 1 if item i_j is annotated with tag t_k, and 0 otherwise.

Through a Bayesian inference, we can derive the posterior distributions of V and T in Eq. (3.9):

$$\begin{aligned}
& p(V, T|D, \sigma_D^2, \sigma_T^2, \sigma_V^2) \\
\propto\ & p(D|V, T, \sigma_D^2) p(V|\sigma_V^2) p(T|\sigma_T^2) \\
=\ & \prod_{j=1}^{n} \prod_{k=1}^{o} [\mathcal{N}(d_{jk}|g(V_j^T T_k), \sigma_D^2)]^{I_{kj}^D} \\
\times\ & \prod_{j=1}^{n} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}) \times \prod_{k=1}^{o} \mathcal{N}(T_k|0, \sigma_T^2 \mathbf{I}). \qquad (3.9)
\end{aligned}$$

3.2.5 A Unified Matrix Factorization for TagRec

Since both users' rating information and users' tagging information can reflect users' judgments about Web contents, we propose a factor analysis approach by utilizing both users' rating information and tagging information based on a unified probabilistic matrix factorization. Specifically, on the one hand, we connect users' rating information with users' tagging information through the shared user latent feature space, and on the other hand, we connect items' received rating information with items' received tagging information through the shared item latent feature space. The shared tag latent feature space is used to represent user-tag tagging information and item-tag tagging information. The graphical model describing the TagRec framework is represented in Fig. 3.1.

According to the graphical model described in Fig. 3.1, we derive the log function of the posterior distributions of TagRec in Eq. (3.10):

$$\begin{aligned}
&\ln p(U, V, T | R, C, D, \sigma_T^2, \sigma_V^2, \sigma_U^2, \sigma_T^2, \sigma_D^2, \sigma_R^2, \sigma_C^2) = \\
&-\frac{1}{2\sigma_R^2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R (r_{ij} - g(U_i^T V_j))^2 \\
&-\frac{1}{2\sigma_C^2}\sum_{i=1}^{m}\sum_{k=1}^{o} I_{ik}^C (c_{ik} - g(U_i^T T_k))^2 \\
&-\frac{1}{2\sigma_D^2}\sum_{j=1}^{n}\sum_{k=1}^{o} I_{jk}^D (d_{jk} - g(V_j^T T_k))^2 \\
&-\frac{1}{2\sigma_U^2}\sum_{i=1}^{m} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{n} V_j^T V_j - \frac{1}{2\sigma_T^2}\sum_{k=1}^{o} T_k^T T_k \\
&-\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R \ln \sigma_R - \sum_{i=1}^{m}\sum_{k=1}^{o} I_{ik}^C \ln \sigma_C - \sum_{j=1}^{n}\sum_{k=1}^{o} I_{jk}^D \ln \sigma_D \\
&-l\sum_{i=1}^{m} \ln \sigma_U - l\sum_{j=1}^{n} \ln \sigma_V - l\sum_{k=1}^{o} \ln \sigma_T + \mathcal{C}, \qquad (3.10)
\end{aligned}$$

where $\mathcal{C}$ is a constant independent of the parameters. We can see the Eq. (3.10) is an unconstrained optimization problem, and maximizing the log-posterior distributions with fixed hyperparameters is equivalent to minimizing the sum-of-squared-errors objective function with quadratic regularized terms in Eq. (3.11):

$$
\begin{aligned}
& E(U, V, T, R, C, D) \\
= & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(r_{ij} - g(U_i^T V_j))^2 \\
+ & \frac{\theta_C}{2}\sum_{i=1}^{m}\sum_{k=1}^{o} I_{ik}^{C}(c_{ik} - g(U_i^T T_k))^2 \\
+ & \frac{\theta_D}{2}\sum_{j=1}^{n}\sum_{k=1}^{o} I_{jk}^{D}(d_{jk} - g(V_j^T T_k))^2 \\
+ & \frac{\theta_U}{2}\sum_{i=1}^{m} U_i^T U_i + \frac{\theta_V}{2}\sum_{j=1}^{n} V_j^T V_j + \frac{\theta_T}{2}\sum_{k=1}^{o} T_k^T T_k, \qquad (3.11)
\end{aligned}
$$

where $\theta_C = \sigma_R^2/\sigma_C^2$, $\theta_D = \sigma_R^2/\sigma_D^2$, $\theta_U = \sigma_R^2/\sigma_U^2$, $\theta_V = \sigma_R^2/\sigma_V^2$, and $\theta_T = \sigma_R^2/\sigma_T^2$. The local minimum can be found by performing the gradient descent on U_i, V_j and T_k, and the derived gradient descent equations are described in Eq. (3.12), Eq. (3.13) and Eq. (3.14) respectively:

$$
\begin{aligned}
\frac{\partial E}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}^{R}(g(U_i^T V_j) - r_{ij})g'(U_i^T V_j)V_j + \theta_U U_i \\
&+ \theta_C \sum_{k=1}^{o} I_{ik}^{C}(g(U_i^T T_k) - c_{ik})g'(U_i^T T_k)T_k, \qquad (3.12)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial E}{\partial V_j} &= \sum_{i=1}^{m} I_{ij}^{R}(g(U_i^T V_j) - r_{ij})g'(U_i^T V_j)U_i + \theta_V V_j \\
&+ \theta_D \sum_{k=1}^{o} I_{jk}^{D}(g(V_j^T T_k) - d_{jk})g'(V_j^T T_k)T_k, \qquad (3.13)
\end{aligned}
$$

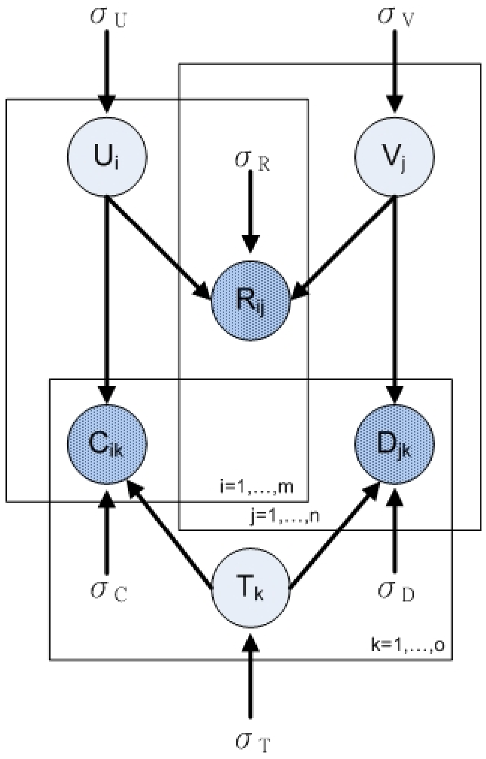


Figure 3.1: Graphical model for TagRec.

$$
\begin{aligned}
\frac{\partial E}{\partial T_k} &= \theta_C \sum_{i=1}^{m} I_{ik}^{C}(g(U_i^T T_k) - c_{ik})g'(U_i^T T_k)U_i + \theta_T T_k \\
&+ \theta_D \sum_{j=1}^{n} I_{jk}^{D}(g(V_j^T T_k) - d_{jk})g'(V_j^T T_k)V_j, \qquad (3.14)
\end{aligned}
$$

where $g'(\cdot)$ is the first-order derivative of the logistic function. We set $\theta_U = \theta_V = \theta_T$ in our experiments in order to reduce the model complexity.

3.2.6 Complexity Analysis

The major computation cost of the gradient descent methods are evaluating objective function E and corresponding gradients on variables. Due to the sparsity of matrices R, C, and D, the complexity of evaluating the objective function in Eq. (3.11) is $\mathcal{O}(n_R l + n_C l + n_D l)$, where n_R, n_C and n_D are the number of non-zero entries in matrices R, C and D respectively, and l is the number of dimensions of latent feature space as shown in Table 3.1. Similarly we can derive the complexities of Eq. (3.12), Eq. (3.13) and Eq. (3.14). Hence, the total complexity for one iteration is $\mathcal{O}(n_R l + n_C l + n_D l)$, which means it is linear with respect to the number of observations in the three sparse matrices. As claimed in [200] the density of available ratings in commercial recommender systems is often less than 1%; therefore, TagRec is efficient and is scalable to large data sets.

3.3 Experimental Analysis

We first ask several research questions intended to give an idea of the highlights of our experimental analysis.

RQ1 How is our approach compared with the baseline methods and the existing state-of-the-art approaches?

RQ2 How do the model parameters θ_C and θ_D affect the prediction accuracies of our approach?

3.3.1 Description of Data Set and Metrics

We use MovieLens $10M/100K$ data set in our experiments. This data set contains $10,000,054$ ratings and $95,580$ tags added to $10,681$ movies by $71,567$ users of the online movie recommender service MovieLens. In order to compare the prediction quality of our method with other methods, we use the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) as the comparison metrics. MAE is defined in Eq. (3.15), and RMSE is defined in Eq. (3.16):

$$MAE = \frac{\sum_{i,j} |r_{i,j} - \hat{r}_{i,j}|}{N}, \tag{3.15}$$

$$RMSE = \sqrt{\frac{\sum_{i,j}(r_{i,j} - \hat{r}_{i,j})^2}{N}}. \tag{3.16}$$

where $r_{i,j}$ denotes the rating user i gave to item j, $\hat{r}_{i,j}$ denotes the predicted rating, and N is the total number of tested ratings.

3.3.2 Performance Comparison

In order to show the prediction performance improvements of TagRec, we compare TagRec with two baseline methods user mean (UMEAN) and item mean (IMEAN). UMEAN is defined in Eq. (3.17) and IMEAN is defined in Eq. (3.18):

$$\hat{r}_{i,j} = \frac{\sum_n r_{i,n}}{N}, \tag{3.17}$$

where $r_{i,n}$ is the observed ratings of user i in the training data and N is the number of observed ratings of user i.

$$\hat{r}_{i,j} = \frac{\sum_m r_{m,j}}{M}, \tag{3.18}$$

where $r_{m,j}$ is the observed ratings of item j in the training data and M is the number of observed ratings of item j. In addition, we ultimately compare our TagRec approach with top-performing recommendation algorithms, including Probabilistic Matrix Factorization (PMF) [193] and Singular Value Decomposition (SVD) [58].

In the comparison, we employ different amount of training data, including 80%, 50%, 30%, 20% and 10%. 80% training data means we randomly select 80% of ratings from the MovieLens $10M/100K$ data set as training data, and leave the remaining 20% as prediction performance testing. The procedure is carried out 5 times independently, and we report the average values in this paper. In the comparison, we set $\theta_U = \theta_V = \theta_T = 0.004$, set $\theta_C = 0.4$ and set $\theta_D = 10$. The MAE results and RMSE results are reported in Table 3.2 and Table 3.3 respectively. From the results, we can see that our TagRec approach consistently outperforms existing algorithms, especially when there is a small amount of training data, which indicates our method performs better under sparse data settings. In addition, it is necessary to notice that in MovieLens $10M/100K$ data set, all the selected users have rated at least 20 movies, but in reality, according to the famous power law distribution phenomenon, in almost all kinds of Web activities most users have rated very few items. Thus, we can see the improvement of TagRec is significant, and this shows the promising future of our TagRec approach.

Table 3.2: MAE comparison with other approaches (a smaller MAE value means a better performance).

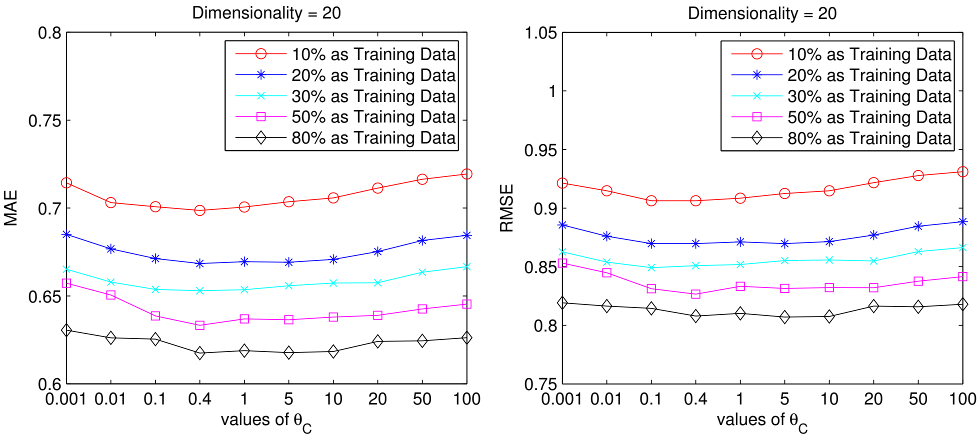
Training	**Baseline Methods**		**Dimensionality = 10**			**Dimensionality = 20**		
Data	**UMEAN**	**IMEAN**	**SVD**	**PMF**	**TagRec**	**SVD**	**PMF**	**TagRec**
80%	0.7686	0.7379	0.6169	0.6162	**0.6159**	0.6167	0.6156	**0.6145**
50%	0.7710	0.7389	0.6376	0.6354	**0.6352**	0.6349	0.6337	**0.6307**
30%	0.7742	0.7399	0.6617	0.6599	**0.6528**	0.6570	0.6569	**0.6494**
20%	0.7803	0.7416	0.6813	0.6811	**0.6664**	0.6776	0.6766	**0.6650**
10%	0.8234	0.7484	0.7315	0.7127	**0.6964**	0.7264	0.7089	**0.6962**

Table 3.3: RMSE comparison with other approaches (a smaller RMSE value means a better performance).

Training	**Baseline Methods**		**Dimensionality = 10**			**Dimensionality = 20**		
Data	**UMEAN**	**IMEAN**	**SVD**	**PMF**	**TagRec**	**SVD**	**PMF**	**TagRec**
80%	0.9779	0.9440	0.8087	0.8078	**0.8077**	0.8054	0.8025	**0.8022**
50%	0.9816	0.9463	0.8330	0.8326	**0.8321**	0.8289	0.8252	**0.8217**
30%	0.9869	0.9505	0.8636	0.8587	**0.8492**	0.8575	0.8553	**0.8450**
20%	1.0008	0.9569	0.8900	0.8824	**0.8659**	0.8857	0.8791	**0.8639**
10%	1.1587	0.9851	0.9703	0.9236	**0.9038**	0.9638	0.9183	**0.9031**

3.3.3 Impact of Parameters θ_C and θ_D

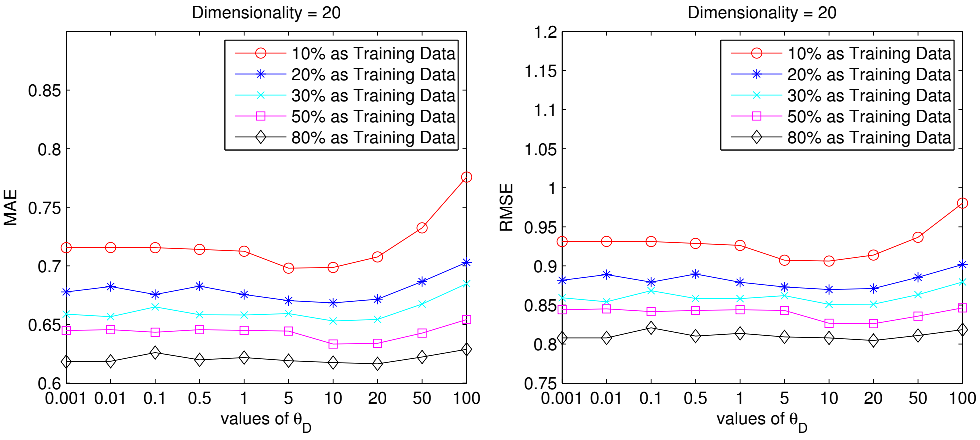
TagRec utilizes both users' rating information and tagging information at the same time to perform the prediction. Specifically, we in-



(a) θ_C, MAE

(b) θ_C, RMSE

Figure 3.2: Dimensionality = 20, impact of parameter θ_C.



(a) θ_D, MAE

(b) θ_D, RMSE

Figure 3.3: Dimensionality = 20, impact of parameter θ_D.

corporate user-item rating matrix, user-tag tagging matrix, and item-tag tagging matrix together based on a unified probabilistic matrix factorization. The parameter θ_C controls the impact of the user-tag tagging matrix, and the parameter θ_D controls the impact of the item-tag tagging matrix. If we set both θ_C and θ_D as 0, it means we only consider users' rating information; if we set both θ_C and θ_D to $+\inf$, it means we only utilize users' tagging information.

We test the impact of these two parameters independently. When we test the impact of parameter θ_C, we set $\theta_U = \theta_V = \theta_T = 0.004$, $\theta_D = 10$, and Fig. 3.2(a) and Fig. 3.2(b) show the results. When we test the impact of parameter θ_D, we set $\theta_U = \theta_V = \theta_T = 0.004$, $\theta_C = 0.4$, and Fig. 3.3(a) and Fig. 3.3(b) present the results. We report results when $dimensionality = 20$, and the results are similar when $dimensionality = 10$. From the results presented in Fig. 3.2 and Fig. 3.3, we can see that both the values of θ_C and θ_D impact the prediction accuracies significantly, and this indicates that utilizing both users' rating information and users' tagging information simultaneously can improve the prediction quality. We further observe that as the value of θ_C or θ_D increases, both the MAE and $RMSE$ first decrease (performances increase); but after θ_C or θ_D is greater than some threshold value, both MAE and $RMSE$ start to increase again (performances decrease). This observation meets our expectation, because only utilizing users' rating information or only utilizing users' tagging information cannot perform better than utilizing rating information and tagging information together. Our approach performs best when $\theta_C \in [0.1, 1]$ and $\theta_D \in [5, 10]$, and the relatively wide range of choosing optimal parameter indicates that the model is easy to train.

3.4 Summary

In this chapter, based on the intuition that both users' rating information and users' tagging information can reflect users' judgments about Web contents, and that tags added to items can represent concepts of items, we propose the TagRec framework, which employs users' rating information and tagging with a unified probabilistic matrix factorization. The experimental results show that the innovative TagRec approach outperforms existing approaches.

□ **End of chapter.**

Chapter 4

User Recommendation via Interest Modeling

4.1 Problem and Motivation

In this chapter, we focus on user interest modeling and user recommendation via interest modeling. Specifically, we employ social tagging systems as a test bed. Social tagging systems have emerged as an effective way for users to annotate and share objects on the Web. However, with the growth of social tagging systems, users are easily overwhelmed by the large amount of data and it is very difficult for users to dig out information that he/she is interested in. Though the tagging system has provided interest-based social network features to enable the user to keep track of other users' tagging activities, there is still no automatic and effective way for the user to discover other users with common interests.

In this chapter, we propose a *User Recommendation* (*UserRec*) framework for user interest modeling and interest-based user recommendation, aiming to boost information sharing among users with similar interests. Our work brings three major contributions to the research community: (1) we propose a tag-graph based community detection method to model the users' personal interests, which are further represented by discrete topic distributions; (2) the similarity values between users' topic distributions are measured by Kullback-

Table 4.1: An example of user-generated tags of a URL.

URL	http://www.nba.com/
Tags of $\mathbf{u_1}$	basketball, nba
Tags of $\mathbf{u_2}$	sports, basketball, nba

Leibler divergence (KL-divergence), and the similarity values are further used to perform interest-based user recommendation; and (3) by analyzing users' roles in a tagging system, we find users' roles in a tagging system are similar to Web pages in the Internet. Experiments on tagging dataset of Web pages (Yahoo! Delicious) show that UserRec outperforms other state-of-the-art recommender system approaches.

4.2 UserRec Framework

4.2.1 User Interest Modeling

A social tagging system consists of users, tags and resources (e.g. URLs, images, or videos), and we define the set of users $U = \{u_i\}_{i=1}^{I}$, the set of tags $T = \{t_k\}_{k=1}^{K}$, and the set of resources $R = \{r_j\}_{j=1}^{J}$. Users can use free-form tags to annotate resources. An annotation of a set of tags to a resource by a user is called a *post* or a *bookmark*. In order to facilitate discussions in the following sections, we define formulas related to *post* as follows:

$$\begin{aligned} R(u) &= \{r_i | r_i \text{ is a resource annotated by } u, r_i \in R\}, \\ S(u) &= \{t_j | t_j \text{ is a tag used by } u, t_j \in T\}, \\ T(u,r) &= \{t_k | t_k \text{ is a tag used by user } u \text{ to annotate the} \\ &\qquad \text{resource } r, t_k \in T\}. \end{aligned}$$

Users in social tagging systems may have many interests, and research efforts have shown that users' interests are reflected in their tagging activities. In addition, patterns of frequent co-occurrences

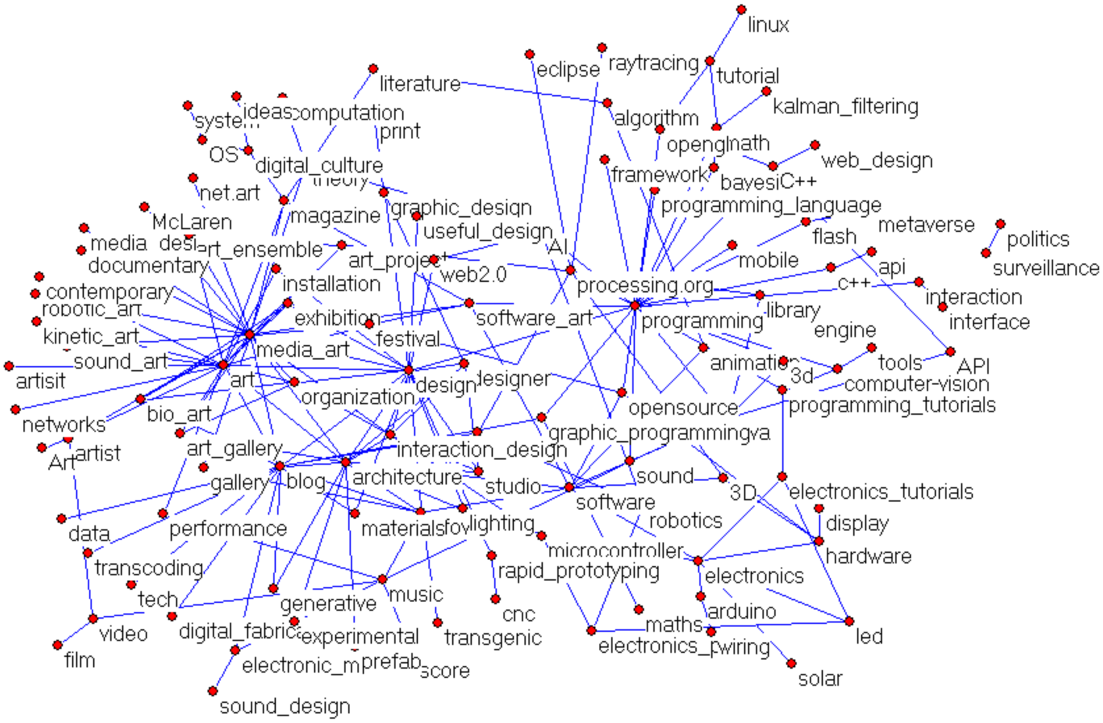


Figure 4.1: Tag graph of one user.

of user tags can be used to characterize and capture users' interests [112]. For example, it is very likely that for two tags t_k and t_m, if $t_k \in T(u_i, r_j)$ and $t_m \in T(u_i, r_j)$, t_k and t_m are semantically-related, and can reflect one kind of this user u_i's interests. Table 4.1 demonstrates one example.

The method for modeling users' interests consists of two stages. In the first stage, we generate an undirected weighted tag-graph for each user. The nodes in the graph are tags used by the user, the weighted edges between two nodes represent the strength of semantic relations between two tags, and the weights are calculated based on the user's tagging activities. Algorithm 1 shows the pseudo code of our method for generating a tag-graph for each user. The intuition of Algorithm 1 is the more often two tags occur together, the more semantically related these two tags are. The generated undirected weighted tag-graph is mapped to an undirected unweighted multigraph based on [152]. Figure 4.1 demonstrates one weighted tag-graph of a user generated by Algorithm 1, and to make the graph clear, the weights are not shown here. Intuitively we can find this user has interests on programming, art, etc. In addition, we can find co-occurrences of tags, such as *art* and *media_art*, *art* and *art_gallery*, can characterize a kind of this user's interests.

Algorithm 1 Generate a tag-graph for user u_i.

```
procedure GenTagGraph(user $u_i$)
Input
$\mathbf{R(u_i)}$, the set of resources annotated by the user $u_i$
$\mathbf{S(u_i)}$, the set of tags used by the user $u_i$
$\forall\mathbf{T(u_i, r_j), where\ r_j \in R(u_i)}$, the set of tags used by the $u_i$ to annotate resource $r_j$
$\mathbf{G(u_i) = (V, E)}$, $V$ are nodes in $G$, $E$ are weighted edges in $G$
$\mathbf{V = \emptyset, E = \emptyset}$
for all $r_j \in R(u_i)$ do
    for all $t_k \in S(u_i)$ do
        for all $t_m \in S(u_i)$ do
            if $t_k \in T(u_i, r_j)$ and $t_m \in T(u_i, r_j)$ then
                if $w(t_k, t_m)$ not exists in $E$ then
                    Add $w(t_k, t_m) = 1$ to $E$
                else
                    $w(t_k, t_m) = w(t_k, t_m) + 1$
                end if
                Add $t_k$ and $t_m$ to $V$ if they not exist in $V$
            end if
        end for
    end for
end for
Output
Tag-graph $\mathbf{G(u_i)}$
```

In the second stage, we adopt a fast greedy algorithm for community discovery in networks [43], which optimizes the modularity, Q, of a network by connecting the two vertices at each step, leading to the largest increase of modularity. For a network of n vertices, after

$n-1$ such joins we are left with a single community, and the algorithm stops. The complexity of the community discovery algorithm is $O(n\log^2 n)$, and n is the number of vertices in the graph. The concept of modularity of a network is widely recognized as a good measure for the strength of the community structure. Modularity is defined in Eq. (4.1):

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j), \tag{4.1}$$

where k_i is degree of node i, and is defined in Eq. (4.2),

$$k_i = \sum_k A_{ik}, \tag{4.2}$$

and A_{ij} is the weight between node i and node j, $\delta(c_i, c_j)$ is 1 if node i and node j belong to the same community after partition; otherwise, $\delta(c_i, c_j)$ is 0. $m = \frac{1}{2}\sum_{ij} A_{ij}$ is the total weights of all edges in this tag-graph. The idea of modularity is that if the fraction of within-community edges is no different from what we would expect for the randomized network, then modularity will be zero. Nonzero values represent deviations from randomness. After detecting communities in previously generated unweighted multigraph, we can find *topics* of a user. A topic, which is represented by a set of tags used by a user in our framework, can show the user's interests. Thus, each community indicates one topic of the user. The set of topics of all the users is named C here. We define all the topics of a user u in Eq. (4.3):

$$UC(u) = \{c^u_m | c^u_m \text{ is a topic of the user u, } c^u_m \in C\}, \tag{4.3}$$

where c^u_m is a topic of the user u, and is defined as follows:

$$\begin{aligned} c^u_m \;=\; & \{t_k | t_k \text{ is a tag belonging to the corresponding} \\ & \quad \text{community of topic } c^u_m \text{ through the} \\ & \quad \text{community detecting algorithm, } t_k \in T\}. \end{aligned} \tag{4.4}$$

Table 4.2: Sample topics of two users.

u_a	sound_art, networks, artist, art, art_gallery kinetic_art, contemporary, artisit, Art
	programming_tutorials, programming_language programming, computer-vision, opengl
u_b	citations, bibliography, Research
	privacy, phishing, myspace, internetsafety cyberbullying, InternetSafety, bullying

Through our proposed two-stage method, we can model users' interests with several topics, which consist one or more tags. Table 4.2 demonstrates sample topics of two users.

4.2.2 Interest-based User Recommendation

Based on the topics of each user generated by our two-stage method for modeling users' interests, we further propose a two-stage method to perform interest-based user recommendation. In the first stage of our interest-based user recommendation method, we represent the topics of each user with a discrete random variable. A probability value is calculated for each topic of a user according to the impact of this topic on the user. Here we introduce how to measure the impact of each topic to a user. In Eq. (4.3) we have defined the formula to express all the topics of a user, and in Eq. (4.4) we have defined the formula to express one topic of a user. $N(t_k, u_i, c_m^{u_i})$ is the number of times tag t_k is used by user u_i, where $t_k \in S(u_i)$, and $t_k \in c_m^u$. We define the impact of a topic $c_m^{u_i}$ to a user u_i in Eq. (4.5):

$$TN(u_i, c_m^{u_i}) = \sum_{t_k \in c_m^{u_i}} N(t_k, u_i, c_m^{u_i}). \tag{4.5}$$

We formulate Eq. (4.5) based on the idea that, if a user uses tags of a topic $c_m^{u_i}$ more often than his or her tags of another topic $c_n^{u_i}$, it is very likely that this user is more interested in the topic $c_m^{u_i}$ than the topic $c_n^{u_i}$. After defining the impact of a topic to a user, we define the total impacts of all the topics on a user in Eq. (4.7). The formula for calculating the probability value of each topic of a user is defined in Eq. (4.6), which shares similar idea with the maximum likelihood estimation method. Through the first stage of our method for performing interest-based user recommendation, we can get users' topic distributions.

$$Pr(u_i, c_m^{u_i}) = \frac{TN(u_i, c_m^{u_i})}{TTN(u_i)}, \tag{4.6}$$

where

$$TTN(u_i) = \sum_{c_m^{u_i} \in UC(u_i)} TN(u_i, c_m^{u_i}). \tag{4.7}$$

In the second stage, we propose a Kullback-Leibler divergence (KL-divergence) based method to calculate the similarity between two users according to their topic distributions. In information theory, the KL-divergence is a measure between two probability distributions. The formula to calculate the similarity value of a user u_j for a user u_i is defined in Eq. (4.8):

$$KL(u_i|u_j) = \sum_{c_m^{u_i} \in UC(u_i)} Pr(u_i, c_m^{u_i}) \log \frac{Pr(u_i, c_m^{u_i})}{Pr(u_j, c_m^{u_j})}. \tag{4.8}$$

Algorithm 2 shows the details of how to calculate the KL-divergence based similarity value of user u_j for user u_i. In line 2 of Algorithm 2, all the tags t_k belong to the same topic $c_m^{u_i}$ are sorted in a descending order according to their used frequencies $N(t_k, u_i, c_m^{u_i})$. The reason for the sorting is that, the more often a tag $t_k, t_k \in c_m^{u_i}$, is used by user u_i to express the topic $c_m^{u_i}$, the more representative this tag t_k is for the topic $c_m^{u_i}$. In other words, different tags may carry different weights to a topic just as different topics may carry different weights to a user. Line 2 to line 7 mean if topic $c_m^{u_i}$ of u_i has a corresponding topic $c_m^{u_j}$ in u_j, the value is calculated and added to the KL-divergence value. Because one topic may contain several tags, the corresponding topic exists if both topics have at least one tag in common. Line 8 to line 12 are used to avoid divide-by-zero problem if no corresponding topic exists, and it is a common way used in calculating the KL-divergence.

Algorithm 2 KL-divergence based similarity measure for user u_j to user u_i.

```
    procedure KL-sim(user $u_i$, user $u_j$)
    Input
    $\forall \mathbf{Pr}(\mathbf{u_i}, \mathbf{c_m^{u_i}}), where\ c_m^{u_i} \in UC(u_i)$
    $\forall \mathbf{Pr}(\mathbf{u_j}, \mathbf{c_m^{u_j}}), where\ c_m^{u_j} \in UC(u_j)$
    $\mathbf{KL}(\mathbf{u_i}|\mathbf{u_j}) = \mathbf{0}$
 1: for all $c_m^{u_i} \in UC(u_i)$ do
 2:     for $t_k \in c_m^{u_i}$ do
 3:         if $t_k \in c_m^{u_j}$ then
 4:             $KL(u_i|u_j) = KL(u_i|u_j)$
 5:             $+Pr(u_i, c_m^{u_i}) \log \frac{Pr(u_i,c_m^{u_i})}{Pr(u_j,c_m^{u_j})}$, BREAK
 6:         end if
 7:     end for
 8:     if $\forall t_k \in c_m^{u_i}, not\ \exists c_m^{u_j}\ \ that\ \ t_k \in c_m^{u_j}$ then
 9:         $KL(u_i|u_j) = KL(u_i|u_j)$
10:         $+Pr(u_i, c_m^{u_i}) \log \frac{Pr(u_i,c_m^{u_i})}{\epsilon}$,
11:         where $\epsilon$ is a very small real value
12:     end if
13: end for
    Output
    $\mathbf{KL}(\mathbf{u_i}|\mathbf{u_j})$
```

4.3 Experimental Analysis

4.3.1 Dataset Description and Analysis

The dataset is crawled from Yahoo! Delicious, and in Yahoo! Delicious, users use free-form tags to annotate URLs that they are interested in. In addition, a user can add other users who share similar interests to their personal *network*. Users are informed the latest interesting resources added by people from his or her *network*. In

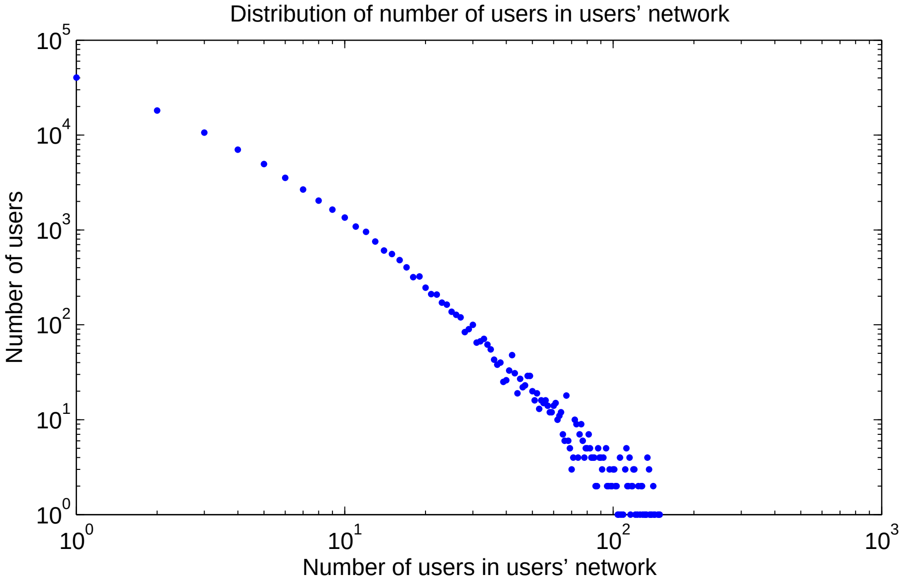


Figure 4.2: Distribution of number of users in users' network.

addition, a user is informed the list of users who have added him or her to their personal *network*, and a list of *fans* appears in this user's profile. In our crawling, we crawl users' bookmarks, and here a bookmark consists of a *user*, a *URL*, and one or several *tags* annotated by this user to this URL. In addition, we crawl users' *network* information and *fans* information. Our crawling lasts one month during year 2009. Table 4.3 shows the statistics of our whole dataset.

Table 4.3: Statistics of the crawled dataset.

Users	**Bookmarks**	**Network***	**Fans****
366,827	49,692,497	425,069	395,415

* This is the total number of users in all users' personal networks.

** This is the total number of fans of all users.

Figure 4.2 shows the distribution of the number of users in a user's network which follows a Power Law distribution.

Figure 4.3 shows the distribution of the number of fans of a user.

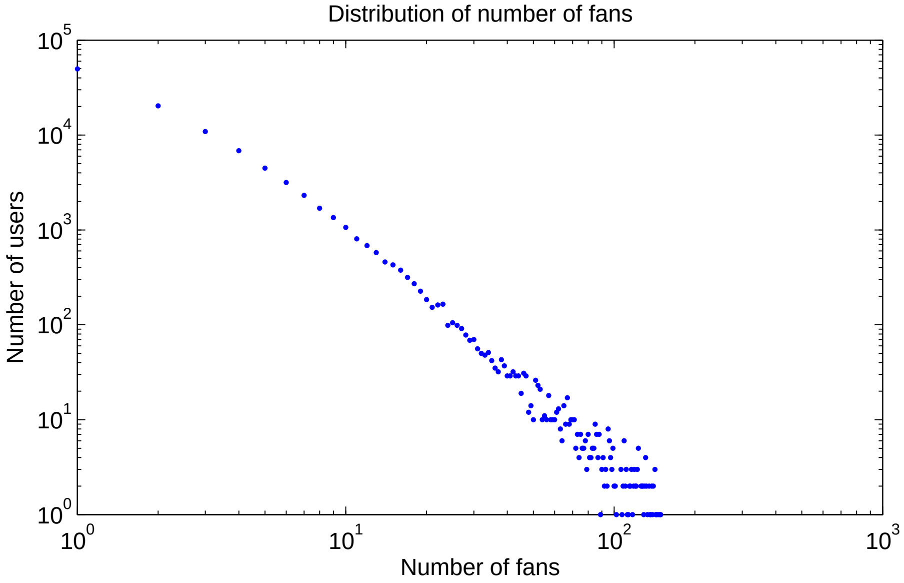


Figure 4.3: Distribution of number of fans.

It is surprised to see that this distribution also follows a Power Law distribution. As we know, the number of fans of a user cannot be determined by the user himself or herself. However, it seems that certain users are well known by other users, and it is interesting to investigate the characteristics of the well known users.

Roughly, *expertise* of a user in Yahoo! Delicious can be interpreted from two aspects: the first is the quality of bookmarked resources and the second is the number of bookmarks. We measure the *expertise* of a user through the second aspect. Figure 4.4 demonstrates the relation between a user's number of bookmarks and his/her number of fans, and we can find there is a positive relationship. The reason why this happens is similar to why the Web portals become very popular and have plenty of visits every day. Note the role of users with extremely large number of bookmarks is very similar to the role of Web portals on the Internet, or called hubs [96].

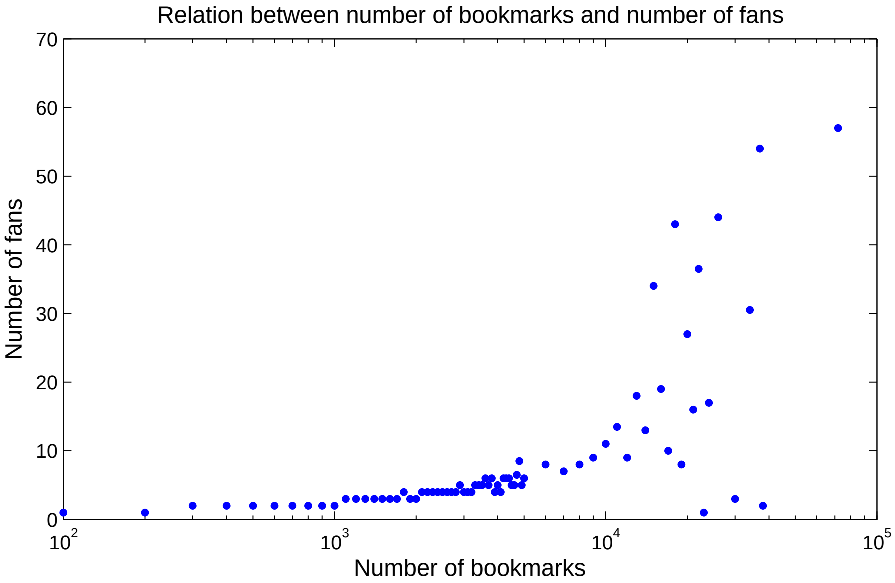


Figure 4.4: Relation between number of bookmarks and number of fans.

4.3.2 Experimental Results

Two research questions are presented to give an idea of the highlights of our experimental analysis:

RQ1 Whether using tags is more effective than using URLs for recommender system approaches?

RQ2 How is our approach compared with the state-of-the-art recommender system approaches?

In order to investigate whether using tags is more effective than using URLs, we employ memory-based approaches and model-based approaches on both URLs and tags, and compare their performances. There are two memory-based approaches we employ, one is the Pearson Correlation Coefficient (PCC) method. The other memory-based approach we compare is the algorithm proposed by [131]. This is an effective PCC-based similarity calculation method with significance weighting, and we refer to it as PCCW. We set the parameter of PCCW to be 30 in our experiments. Two top-performing model-based recommendation algorithms are also employed, including Probabilistic Matrix Factorization (PMF) proposed by [193], and Singular Value Decomposition (SVD) proposed by [58]. Both PMF and SVD employ matrix factorization approach to learn high quality low-dimensional feature matrices. After deriving the latent feature matrices, we still need to use memory-based approaches on derived latent feature matrices to perform the user recommendation task, and we employ both PCC and PCCW on latent feature matrices of SVD and PMF. We refer to them as SVD-PCC, SVD-PCCW, PMF-PCC, and PMF-PCCW respectively. We tune the dimension of latent matrices and set the optimal dimension value 10, and use five-folder cross-validation to learn the latent matrices for SVD and PMF.

It is shown that spreading interests within the *network* of Yahoo! Delicious users contribute a lot to the increase of popularity of resources [237]. Thus, by crawling users' *network*, for each user in the test data, we consider users in his/her network share similar interests with him/her. In other words, users in a user's network is considered as relevant results in the user recommendation task. We randomly sample 400 users whose number of users in their network is between 3 and 10, and further collect all the users in these 400 users' network resulting in $2,376$ users in total. Then we crawl all these $2,376$ users' bookmarks, and there are total $1,190,762$ unique URLs and $139,707$ unique tags.

We adopt four well known metrics that capture different aspects of the performance for the evaluation of the task, namely Precision at rank n (P@n), Precision at rank R(P@R), Mean Average Precision (MAP) and Bpref [32]. In addition, we propose three novel metrics to help further evaluate the effectiveness of the proposed UserRec framework, namely Mean Multi-valued Reciprocal Rank (MMVRR), Top-K accuracy and Top-K recall. Multi-valued Reciprocal Rank (MVRR) is revised from the measure *reciprocal rank*. In our experimental scenario, the input of each measure is a user, and there are several relevant results for each user. We de-

Table 4.4: Comparison with approaches based on URLs (a larger value means a better performance for each metric).

Metrics	Memory-Based Models		Model-Based Models				UserRec
	PCC	PCCW	SVD+ PCC	SVD+ PCCW	PMF+ PCC	PMF+ PCCW	
P@R	0.0717	0.1490	0.0886	0.0907	0.1136	0.1322	**0.3272**
MAP	0.1049	0.1874	0.1218	0.1245	0.1491	0.1745	**0.3752**
Bpref	0.0465	0.1148	0.0568	0.0582	0.0765	0.1029	**0.2913**
MMVRR	0.0626	0.1154	0.0710	0.0736	0.0858	0.1088	**0.2345**

Table 4.5: Comparison with approaches based on tags (a larger value means a better performance for each metric).

Metrics	Memory-Based Models		Model-Based Models				UserRec
	PCC	PCCW	SVD+ PCC	SVD+ PCCW	PMF+ PCC	PMF+ PCCW	
P@R	0.1495	0.3168	0.1540	0.2042	0.1875	0.2084	**0.3272**
MAP	0.1816	0.3444	0.1898	0.2469	0.2084	0.2440	**0.3752**
Bpref	0.1132	0.2395	0.1170	0.1479	0.1376	0.1707	**0.2913**
MMVRR	0.1129	0.1943	0.1151	0.1397	0.1300	0.1550	**0.2345**

fine $MVRR(u) = \sum_{i=1}^{N} \frac{1}{u_{r_i}}/N$, where u_{r_i} is the rank of a relevant result of user u, and N is total number of relevant results of user u. Mean Multi-valued Reciprocal Rank (MMVRR) is the mean value of MVRR in the test set. Top-K accuracy measures percentage of users who actually add at least one of the Top K-th recommended user in his/her network. Top-K recall measures percentage of people in users' network covered by top K recommended users.

Table 4.4 demonstrates the results of metrics Precision@R, MAP, Bpref, and MMVRR of our method and other approaches when employing URLs.

Table 4.5 shows the results of these metrics of our method and other approaches when employing tags.

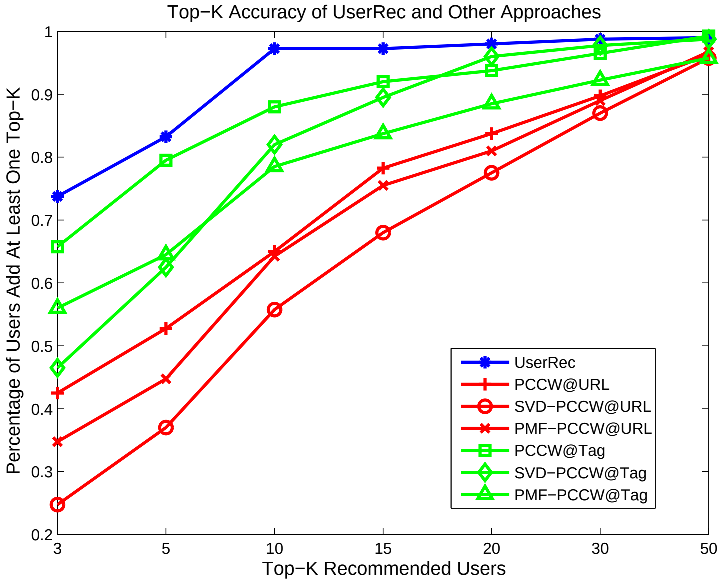


Figure 4.5: Top-K accuracy of different approaches.

From Table 4.4 and Table 4.5, we can see that the proposed UserRec consistently outperforms other approaches on all these metrics. In addition, comparing results in Table 4.5 with results in Table 4.4, we can see that the same approaches achieve better performances when employing tags' information than when employing URLs' information, and this further confirms that tags can capture users' interests.

In order to further compare the effectiveness of the proposed method with state-of-the-art approaches, and to further investigate whether employing tags can achieve better performances than employing URLs, we show Top-K accuracy, Top-K recall and Precision@N results of our method, and results of PCCW, SVD-PCCW, PMF-PCCW when employing on tags and URLs respectively.

Figure 4.5 shows Top-K accuracy of different approaches.

Figure 4.6 shows Top-K recall of different approaches.

Figure 4.7 shows Precision@N of different approaches.

From the results of Fig. 4.5, Fig. 4.6 and Fig. 4.7, we can see the proposed UserRec method still outperforms other approaches

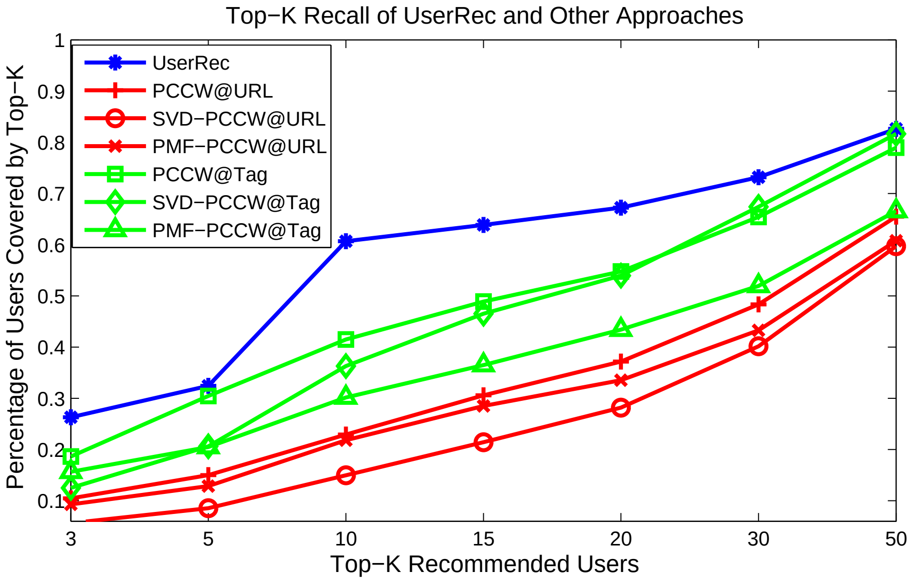


Figure 4.6: Top-K recall of different approaches.

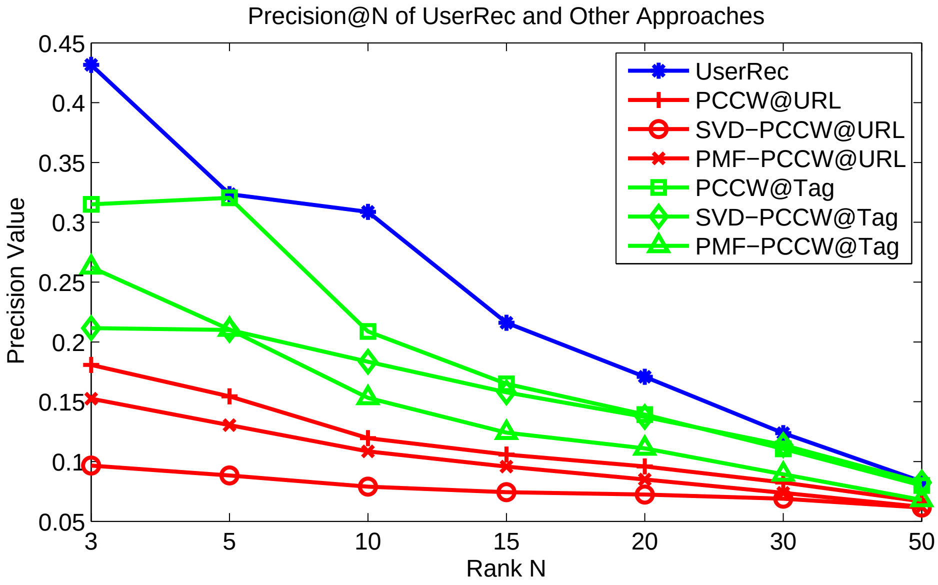


Figure 4.7: Precision@N of different approaches.

in each metric, which is quite encouraging. In addition, we can find that the results of PCCW@Tag, SVD-PCCW@Tag, and PMF-PCCW@Tag are better than PCCW@URL, SVD-PCCW@URL and PMF-PCCW@URL respectively. From these three metrics, we can again confirm that tags are quite good resources to characterize users' interests.

4.4 Summary

In this chapter, we propose an effective framework for users' interest modeling and interest-based user recommendation in social tagging systems, which can help information sharing among users with similar interests. Specifically, we analyze the *network* and *fans* properties, and we observe an interesting finding that the role of users have similar properties with Web pages on the Internet. Experiments on a real world dataset show encouraging results of UserRec compared with the state-of-the-art recommendation algorithms. In addition, experimental results also confirm that tags are good at capturing users' interests.

□ **End of chapter.**

Chapter 5

Item Suggestion with Semantic Analysis

5.1 Problem and Motivation

In this chapter, we focus on item suggestion with semantic analysis. Specifically, we focus on question suggestion in social media. Social media systems with Q&A functionalities have accumulated large archives of questions and answers. Two representative types are online forums and community-based Q&A services. A travel forum TripAdvisor has 45 million reviews[1]. 10 questions and answers are posted per second in Yahoo! Answers[2]. About 218 million questions have been solved in Baidu Knows[3]. In this chapter, we address the problem of Question Suggestion, which targets at suggesting questions that are semantically related to a queried question. Existing bag-of-words approaches suffer from the shortcoming that they could not bridge the lexical chasm between semantically related questions. Therefore, we present a new framework to suggest questions, and propose the Topic-enhanced Translation-based Language Model (TopicTRLM), which fuses both the lexical and latent semantic knowledge. This fusing enables TopicTRLM to find

[1] http://www.prnewswire.com/news-releases/tripadvisor-grows-and-grows-and-grows-119678844.html

[2] http://yanswersblog.com/index.php/archives/2010/05/03/1-billion-answers-served/

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

semantically related questions to a given question even when there is little word overlap. Moreover, to incorporate the answer information into the model to make the model more complete, we also propose the Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A). The answer information is utilized under a language modeling framework. Extensive experiments have been conducted with real world data sets from online forums and community-based Q&A services. Experimental results indicate our approach is very effective and outperforms other popular methods in several metrics.

5.2 Question Suggestion Framework

In this section, we present our approach of learning to suggest questions in online forums and community-based Q&A services. Specifically, we start by introducing the method of question detection in online forums, then we explain topic-enhanced translation-based language model (TopicTRLM) for measuring semantic relatedness of two questions, followed by learning word translation probabilities in online forums. To make this article self-contained, we briefly introduce Latent Dirichlet Allocation (LDA). Finally, we introduce topic-enhanced translation-based language model with answer ensemble (TopicTRLM-A) for measuring questions' relatedness in community-based Q&A services.

5.2.1 Question Suggestion in Online Forums

We propose an effective question suggestion framework in online forums. The framework in Fig. 5.1 consists of three major steps: (1) detecting questions in forum threads; (2) learning word translation probabilities from questions in forum threads; (3) calculating semantic relatedness between a queried question and a candidate question using Topic-enhanced Translation-based Language Model

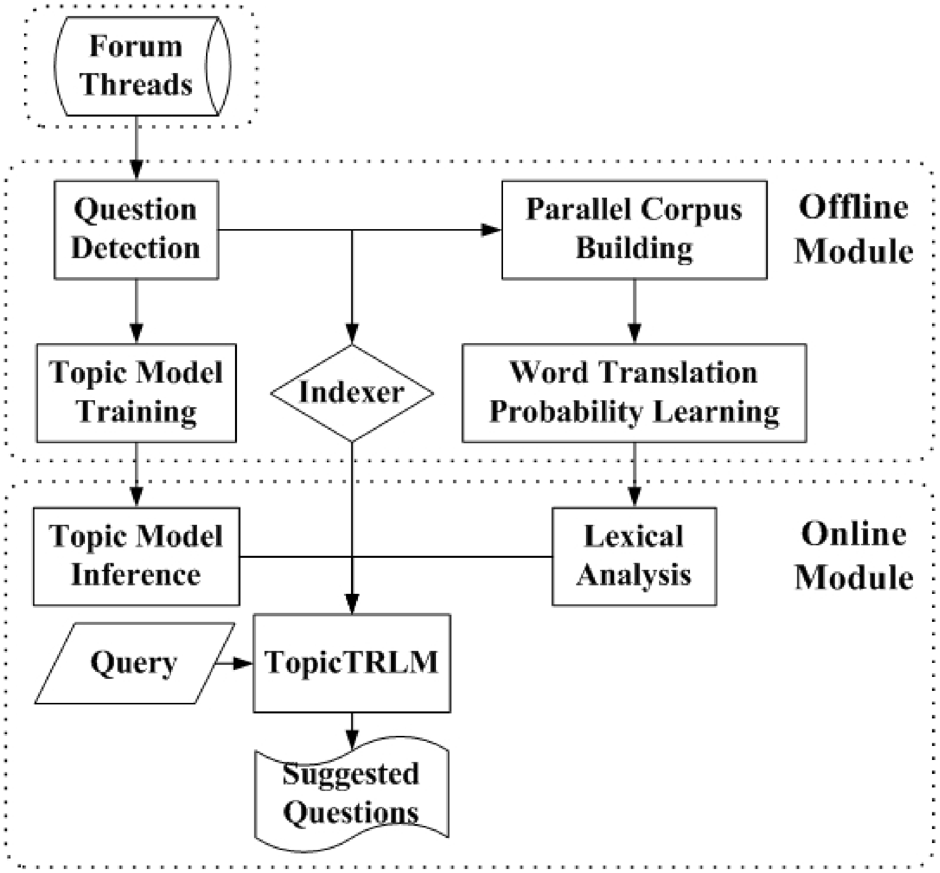


Figure 5.1: System framework of question suggestion in online forum.

(TopicTRLM). In the proposed framework, we utilize interactive nature of forum threads to learn word translation probabilities, and fuse both the lexical and latent semantic knowledge to calculate the semantic relatedness between two questions.

Question Detection in Online Forums

Questions are usually the focus of forum discussions and a natural means of resolving issues. But simple rules, such as question mark and 5W1H (who, where, when, why, what and how) words, are inadequate for detecting questions in online forums. In this paper, we adopt the method proposed by Cong et al. [44] for question detection since that method can achieve both high recall and high precision. Specifically, Labeled Sequential Patterns (LSPs) from both questions and non-questions are extracted as features to build a classifier for question detection.

Definition 1: A labeled sequential pattern p is in the form of $S \rightarrow l$, where S is a sequence of items, and l is a class label. Denoting $\mathcal{IS}$ as the item set, and $\mathcal{LS}$ as the class label set, we can represent each tuple in a sequence database $\mathcal{SD}$ as a list of items in $\mathcal{IS}$ and a class label in $\mathcal{LS}$. A sequence $s_2 =< b_1, \ldots, b_n >$ contains a sequence $s_1 =< a_1, \ldots, a_m >$ if there exists integers $i_1, \ldots, i_m$ such that $1 \leq i_1 < \ldots < i_m \leq n$ and $a_j = b_{ij}$, for all $j = 1, \ldots, m$. In addition, the distance between the two adjacent items b_{ij} and b_{ij+1} in s_2 need to be less than a threshold. In addition, a LSP p_1 is contained by p_2 if the sequence in p_1 is contained by the sequence in p_2 and they have the same class label.

Definition 2: Support of a LSP p, denoted as $sup(p)$, is the percentage of tuples in a sequence database $\mathcal{SD}$ that contain the p. $sup(p)$ measures the generality of the pattern p.

Definition 3: Confidence of a LSP p, denoted as $conf(p)$, is calculated by $sup(p)/sup(p.S)$. $conf(p)$ measures predictive ability of p.

Topic-Enhanced Translation-based Language Model

Two types of methods are typically used to represent the content of text documents. One is the bag-of-words representation, which means that words are assumed to occur independently. A bag-of-words model is a fine-grained representation of a text document. The other method to represent text documents is topic model. Topic model assigns a set of latent topic distributions to each word by capturing important relationships between words. Comparing with bag-of-words representation, topic model is a coarse-grained representation for documents.

Suggested questions should be semantically related to the queried question, and they should explore different aspects of a discussion topic with respect to the queried question. Fine-grained bag-of-words representation of question would contribute to finding lexically similar questions, and topic model representation would contribute to finding semantically related questions. To achieve the goal of adopting both bag-of-words and topic model representations, we propose the TopicTRLM model. It fuses the latent topic information with lexical information to measure the semantic relatedness between two questions systematically. Specifically, we employ the Translation-based Language Model (TRLM) to measure the semantic relatedness of bag-of-words representations of two questions and employ Latent Dirichlet Allocation (LDA) to calculate the latent topics' similarities between two questions.

Equation 5.1 shows TopicTRLM approach to calculate the semantic relatedness of a queried question and a candidate question:

$$\begin{aligned} P(q|D) &= \prod_{w \in q} P(w|D), \qquad (5.1) \\ P(w|D) &= \gamma \times P_{trlm}(w|D) + (1-\gamma) P_{lda}(w|D), \end{aligned}$$

where q is the queried question, D is a candidate question, w is a query term in q, $P_{trlm}(w|D)$ is the TRLM score, and $P_{lda}(w|D)$ is the LDA score. Equation 5.1 employs Jelinek-Mercer smoothing [250] to fuse the TRLM score with LDA score, and γ is the parameter to balance the weights of bag-of-words representation and topic-model representation. A larger γ means that we would like to find more lexically related questions for the queried question; a smaller γ would emphasize more on two questions' latent topic distributions' similarity. When we set $\gamma = 0$, TopicTRLM only employs latent topic analysis, and when we set $\gamma = 1$, TopicTRLM only employs lexical analysis. Thus, TopicTRLM is a generalization of both lexical analysis and latent topic analysis in the question suggestion task. Equation 5.2 describes TRLM which employs

Dirichlet smoothing:

$$\begin{aligned} P_{trlm}(w|D) &= \frac{|D|}{|D|+\lambda} P_{mx}(w|D) + \frac{\lambda}{|D|+\lambda} P_{mle}(w|C), \quad (5.2) \\ P_{mx}(w|D) &= \delta P_{mle}(w|D) + (1-\delta) \sum_{t \in D} T(w|t) P_{mle}(t|D), \end{aligned}$$

where $|D|$ is the length of the candidate question, C is the question collection extracted from the forum posts. λ is the Dirichlet smoothing parameter to balance the collection smoothing and empirical data. If we increase λ, then we would rely more on smoothing. Dirichlet smoothing has the advantage that for longer candidate questions. Its smoothing effect would be smaller. δ is the parameter to balance between language model and translation model. A larger δ would have the effect to retrieve lexically similar questions. A smaller δ would have the effect to retrieve lexically related questions. $T(w|t)$ is the translation probability from source word t to target word w, $P_{mle}(\cdot)$ is the maximum likelihood estimation. An important part of TRLM is to learn the word-to-word translation probabilities $T(w|t)$, which would be discussed later. Equation 5.3 describes employing LDA to calculate the similarity between a query term w and a candidate D:

$$P_{lda}(w|D) = \sum_{z=1}^{K} P(w|z) P(z|D), \quad (5.3)$$

where K is the number of latent topics, and z is a latent topic.

Learning Translation Probability in Online Forums

Learning word-to-word translation probabilities is the most essential part in TRLM. IBM model 1 [31] is employed to learn the translation probabilities, and a monolingual parallel corpus is needed. The construction of the parallel corpus should be tailored to the specific task.

To find similar questions, three kinds of approaches are employed previously to build parallel corpus:

1. Question and question pairs are considered as a parallel corpus if their answers are similar [87, 86].
2. Question and answer pairs are considered as a parallel corpus [244].
3. Question and its manually labeled question reformulation pairs are considered as a parallel corpus [23].

However, neither of above three methods is suitable to build the parallel corpus for the question suggestion task in forums. The reason is that the presence of spam within the discussion forum would make all questions subjected to the same spam appear equivalent. To build a parallel corpus for learning word-to-word translation probabilities for question suggestion, we turn to investigating the properties of forum discussions. Because questions are usually the focus of forum discussions and a natural means of resolving issues, questions posted by a thread starter during the discussion are very likely to explore different aspects of a topic. It is very likely that these questions are semantically related. Thus, we propose to utilize these semantically related questions posted by the thread starter in each thread to build the parallel corpus. The procedure of generating a parallel corpus of related questions from forums is as follows:

1. Extract questions posted by the thread starter in a thread, and create a question pool Q.
2. Construct question-question pairs by enumerating all possible combinations of question pairs in the Q.
3. Repeat above two steps for each forum thread.
4. Build the parallel corpus by aggregating all question-question pairs constructed from each forum thread.

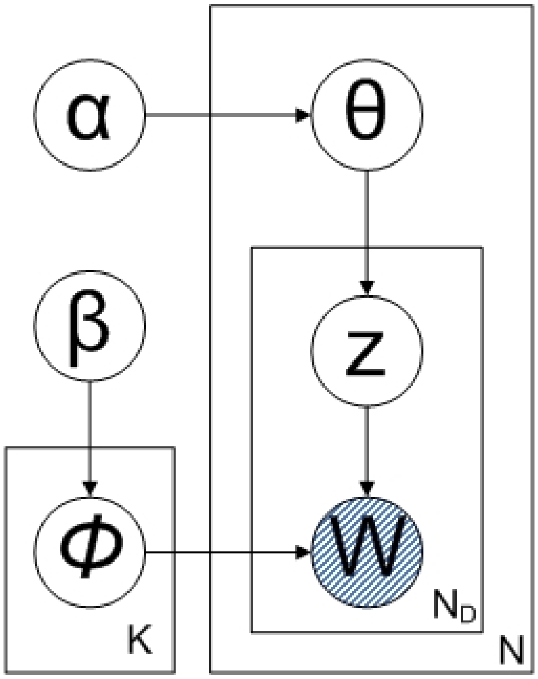


Figure 5.2: Graphical model of LDA. N is the number of documents; N_D is the number of words in document D; K is the number of topics.

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [27], as a topic model method that possesses fully generative semantics, has attracted a lot of interests in the machine learning field. The graphical model of LDA is shown in Fig. 5.2:

The process of generating a corpus in the smoothed LDA is as follows: (1) pick a multinomial distribution φ_z for each topic z from a Dirichlet distribution with parameter β; (2) pick a multinomial distribution θ_D from a Dirichlet distribution with parameter α for each question D; (3) pick a topic $z \in \{1, \ldots, K\}$ from the multinomial distribution θ_D for each word token w in question D; (4) pick word w from the multinomial distribution φ_z.

We calculate the semantic relatedness between a query word w

and a candidate question D as follows:

$$P_{lda}(w|D,\theta,\varphi)=\sum_{z=1}^{K}P(w|z,\varphi)P(z|\theta,D), \tag{5.4}$$

where θ and φ are the posterior. We employ Gibbs sampling to directly obtain the approximation of θ and φ because the LDA model is quite complex and cannot be solved by exact inference [66]. In a Gibbs sample, φ is approximated with $(n^{(w_i)}_{-i,j}+\beta_{w_i})/\sum_{v=1}^{V}(n^{(v)}_{-i,j}+\beta_v)$, and θ is approximated with $(n^{(D_i)}_{-i,j}+\alpha_{z_i})/\sum_{m=1}^{M}(n^{(D_i)}_{-i,m}+\alpha_m)$ after a certain number of iterations being accomplished. $n^{(w_i)}_{-i,j}$ is the number of instances of word w_i assigned to topic $z=j$, not including the current token. α and β are hyper-parameters that determine how heavily this empirical distribution is smoothed. $n^{(D_i)}_{-i,j}$ is the number of words in document D_i assigned to topic $z=j$, not including the current token. The total number of words assigned to topic $z=j$ is $\sum_{v=1}^{V}n^{(v)}_{-i,j}$. The total number of words in document D not including the current one is $\sum_{m=1}^{M}n^{(D_i)}_{-i,m}$. Based on these derivations, we have following mathematical equation:

$$P_{lda}(w|D)=\sum_{z=1}^{K}\frac{n^{w_i}_{-i,j}+\beta_{w_i}}{\sum_{v=1}^{V}(n^{(v)}_{-i,j}+\beta_v)}\times\frac{n^{D_i}_{-i,j}+\alpha_{z_i}}{\sum_{m=1}^{M}(n^{(D_i)}_{-i,m}+\alpha_m)}. \tag{5.5}$$

5.2.2 Question Suggestion in Community-based Q&A Services

We propose a question suggestion framework in community-based Q&A. The framework in Fig. 5.3 consists of two modules: offline module and online module. In the offline module, we utilize questions and their best answers to learn word translation probabilities

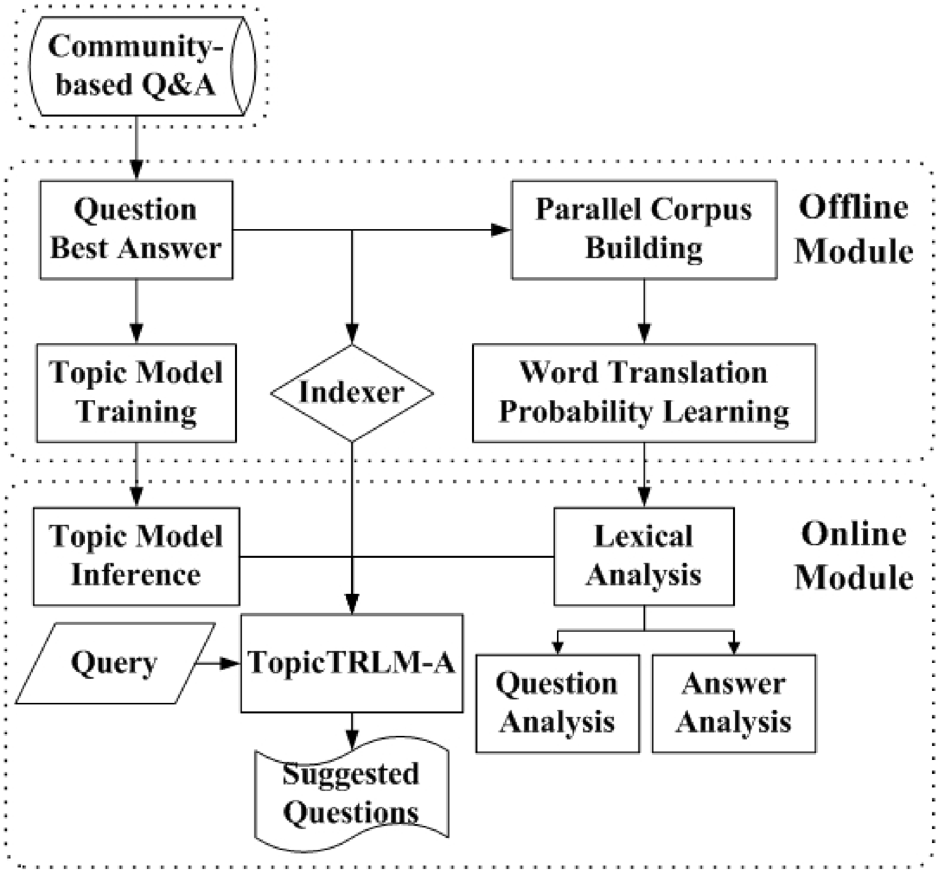
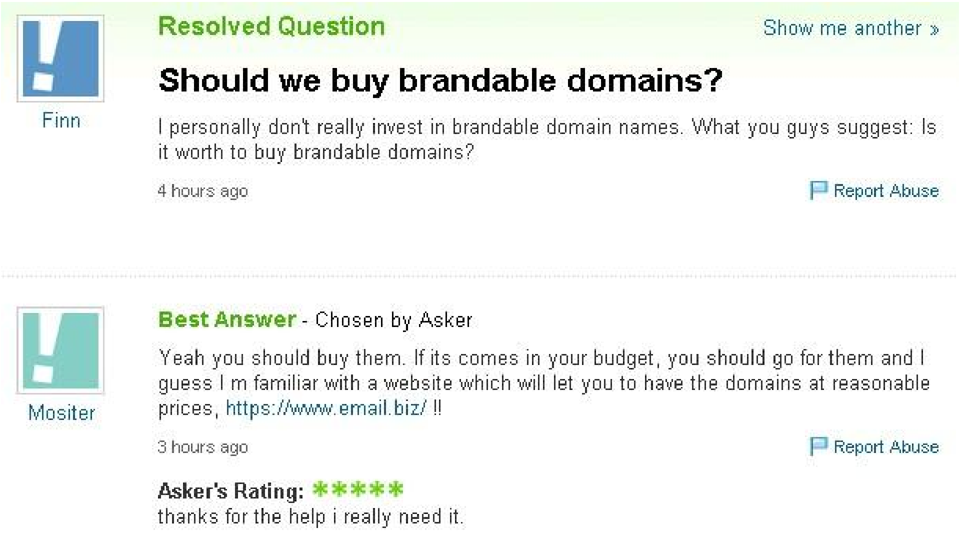


Figure 5.3: System framework of question suggestion in community-based Q&A.

and train topic models. In the online module, we compute semantic relatedness between a queried question and a candidate question using Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A).

Topic-Enhanced Translation-based Language Model with Answer Ensemble

Community-based Q&A services, such as Yahoo! Answers, are question-centric, in which users are socially interacting by engaging in multiple activities around a specific question. Thus, we do not need to perform question detection as we do in online forums. When a user asks a new question, he/she also assigns it to a specific category, within a predefined hierarchy of categories, which should



Resolved Question

Should we buy brandable domains?

Finn

I personally don't really invest in brandable domain names. What you guys suggest: Is it worth to buy brandable domains?

4 hours ago

Best Answer - Chosen by Asker

Mositer

Yeah you should buy them. If its comes in your budget, you should go for them and I guess I m familiar with a website which will let you to have the domains at reasonable prices, https://www.email.biz/ !!

3 hours ago

Asker's Rating: *****

thanks for the help i really need it.

Figure 5.4: Example of a resolved question in Yahoo! Answers with question title, question detail and best answer.

best match the question's general topic. The new question remains "open" for four days with an option for extension, or for less if the asker chose a best answer within this period. Registered users may answer a question as long it is "open". If after this time period the question remains unresolved, its status changes from "open" to "in-voting", in which users can only vote for a best answer till a clear winner arises. Thus, a best answer is always available for a resolved question either chosen by the asker or voted by communities. Most community-based Q&A services allow users to write a "question title" to describe their questions in one sentence, and write a "question detail" to elaborate their question in detail. An example of a resolved question in Yahoo! Answers is shown in Fig. 5.4. In it, we can find that a question detail is provided along with the question title to help elaborate the background, and a best answer is chosen by the asker for this resolved question.

Since the best answer for each resolved question in community-based Q&A services is always readily available, we propose to incorporate it into our model, and propose the topic-enhanced translation-based language model with answer ensemble (TopicTRLM-A). The intuition is that the best answer of a question could also explain the semantic meaning of the question. Thus, when we measure the semantic relatedness of a queried question and a candidate question, we also consider the semantic relatedness between the queried question and the best answer of a candidate question. The mathematical equation of TopicTRLM-A is shown in Eq. 5.6:

$$\begin{aligned} P(q|(Q,A)) &= \prod_{w \in q} P(w|(Q,A)), \\ P(w|(Q,A)) &= \epsilon P_{trlm}(w|(Q,A)) + (1-\epsilon) P_{lda}(w|Q), \end{aligned} \tag{5.6}$$

where q is a queried question, (Q, A) is a candidate question with its best answer, w is a word in the queried question. $P_{trlm}(w|(Q, A))$ is the lexical score, and $P_{lda}(w|Q)$ is the latent semantic score. ϵ is a parameter to balance lexical score and latent semantic score. If we set a large ϵ, we would reply more on lexical score, if we set a small ϵ, we would reply more on latent semantic score. Equation 5.7 presents the details of lexical score calculation:

$$\begin{aligned} P_{trlm}(w|(Q,A)) &= \frac{|(Q,A)|}{|(Q,A)|+\lambda} P_{mx}(w|(Q,A)) \\ &+ \frac{\lambda}{|(Q,A)|+\lambda} P_{mle}(w|C), \\ P_{mx}(w|(Q,A)) &= \eta P_{mle}(w|Q) + \theta \sum_{t \in Q} T(w|t) P_{mle}(t|Q) + \mu P_{mle}(w|A), \end{aligned} \tag{5.7}$$

where we employ the Dirichlet smoothing between the candidate question and the collection. $|(Q, A)|$ is the length of a candidate question with its best answer. If we set a large λ, we would have a larger smoothing effect. We employ translation-based language model on the question part, and incorporate the best answer using language model. η, θ and μ are parameters to represent weights on each part, where $\eta + \theta + \mu = 1$.

Learning Translation Probability in Community-based Q&A Services

Learning word translation probability in community-based Q&A services is an important part of TopicTRLM-A model. Different from online forums, there is usually only one question in each thread in community-based Q&A services. After observing the real world data, we find the question detail is usually a reformulation of the corresponding question title. Thus, we aggregate question title and question detail as a monolingual parallel corpus, and employ IBM model 1 to learn the word translation probabilities.

5.3 Experiments And Results

In this section we describe the experiments we conducted to test our novel models. We conducted experiments on both TripAdvisor and Yahoo! Answers, to demonstrate the effectiveness of the proposed models in online forums and community-based Q&A services separately. TripAdvisor is a popular forum attracts many discussions on travel related topics, and Yahoo! Answers is one of the most renowned community-based Q&A services. In each part, we first describe our experimental setup, including the dataset we used, metrics we employed, and methods we compared. Then, we present the results of our experiments and analysis that shed more light on the performance of our models.

5.3.1 Experiments in Online Forums

We consider the question suggestion task as a retrieval task in our experiments. We aim to address three research questions on question suggestion in online forums:

RQ1: How effective is the proposed method to learn the word-to-word translation probabilities in online forums?

RQ2: How is TopicTRLM compared with other approaches on labeled questions in question suggestion task in online forums?

RQ3: How is TopicTRLM compared with other approaches on the joint probability distributions' similarity of topics with ground truth?

Methods: To evaluate the performance of the proposed methods, we compared the proposed algorithms with alternative approaches. Specifically, we compared our method topic-enhanced translation-based language model (TopicTRLM) with LDA [27], query likelihood language model using Dirichlet smoothing (QL) [250], translation model (TR) [87, 86], and the state-of-the-art question search method translation-based language model (TRLM) [244].

Data set: TripAdvisor is a popular online forum that attracts a large number of discussions about hotels, traveler guides, etc. TripAdvisor forum consists of a large number of threads, which contain posts from thread starters and other participants. For evaluation purpose, we crawled data from the travel forum TripAdvisor. The crawling process was conducted from the thread level. We employed the same settings with Cong et al. [44] to mine LSPs, and the classification-based question detection method was reported to score 97.8% in Precision, 97.0% in Recall, and 97.4% in F1-score.

After employing the question detection method in crawled data, we randomly sampled 300 questions, we removed questions that are not comprehensible, e.g., "What to see?" is not a comprehensible question; while "How is the Orange Beach in Alabama?" is a comprehensible question. Finally we obtained 268 questions. We used the unigram language model to represent questions, and applied IBM model 1 to learn unigram to unigram translation probabilities. We deployed Porter Stemmer [171] to stem question words. We adopted the stop word list used by SMART system [33], but 5W1H words were removed from the stop word list. For each model, the top 20 retrieval results were kept. We used pooling [136] to put results from different models for one query together for annotation, and all models were used in the pooling process. If a returned result was considered as semantically related to the queried question,

Table 5.1: Statistics of post level activities of thread starter (TS).

#Threads	**#Threads that have replied posts from TS**	**Avg.# replied posts from TS**
1,412,141	566,256	1.9

it was labeled with "relevant"; otherwise, it was labeled with "irrelevant". Two assessors were involved in the initial labeling process. If two assessors had different opinions on a decision, a third assessor was asked to make a final decision. The kappa statistics between two assessors was 0.74. This test set was referred to as "TST_LABEL".

We tried to create a reasonable ground truth data without involving laborious manual labeling. Thus, we assumed that questions posted by the same user in a thread were related. We built the unlabeled testing data set by randomly selecting threads until there were $10,000$ threads that contained at least two questions posted by thread starters. The first question in each thread was treated as the queried question. This test set was referred to as "TST_UNLABEL".

The remaining questions, referred to as "TRAIN_SET", were used in three purposes: (1) building parallel corpus to learn the word-to-word translation probabilities, (2) LDA training data, and (3) question repository to retrieve questions to offer question suggestion service. TRAIN_SET contained $1,976,522$ questions extracted from $971,859$ threads. We conducted a detailed analysis on the TRAIN_SET to acquire a deeper understanding of the forum activities.

This paper leveraged thread starters' activities in forums, so we first conducted a post level analysis on thread starters' activities. The statistics is shown in Table 5.1.

From Table 5.1, we can see thread starters replied on average 1.9 posts to the thread he or she initiated, and this indicates our expectation that forum discussions are quite interactive. We also plotted the distribution of replied posts from thread starter in Fig. 5.5, and this distribution follows a power law distribution.

We also conducted a question level analysis on thread starters'

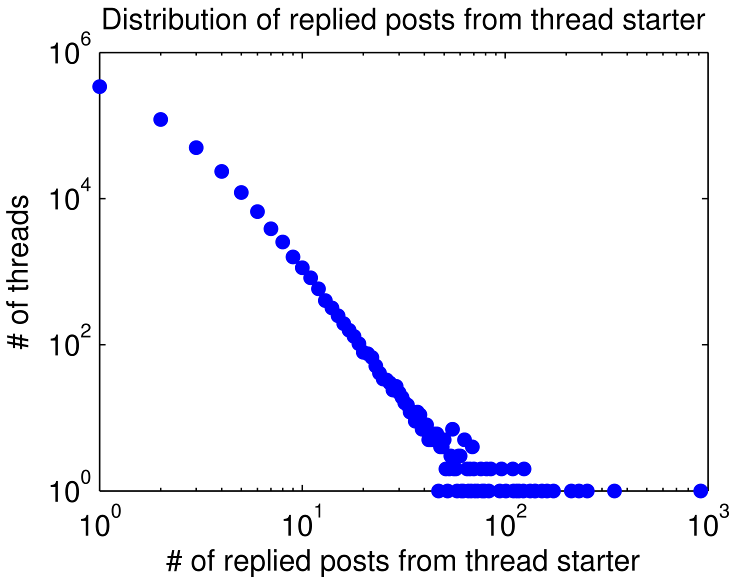


Figure 5.5: Post level distribution of thread starters' activity.

Table 5.2: Statistics of question level activities of thread starter (TS).

#Threads	**#Threads TSs' posts contain questions**	**Avg.# questions in TSs' posts**
1,412,141	971,859	2.0

activities. Table 5.2 presents statistics of question level activities of thread starter. We found over 68.8% thread starters asked on average 2 questions in each thread. These findings supported our motivation that question is a focus of forum discussions, and forum data is an ideal source to train the proposed model for question suggestion.

Figure 5.6 depicts a view of distribution of questions in thread starter's posts. We can see this distribution also follows a power law distribution.

We used 5 as the threshold for sequential pattern mining in question detection. In this paper, we mined LSPs by considering both minimum support threshold and minimum confidence threshold. We empirically set minimum support at 0.5% and minimum confidence at 85% using a development corpus. Each discovered LSP formed a binary feature as the input for a classification model. We used a

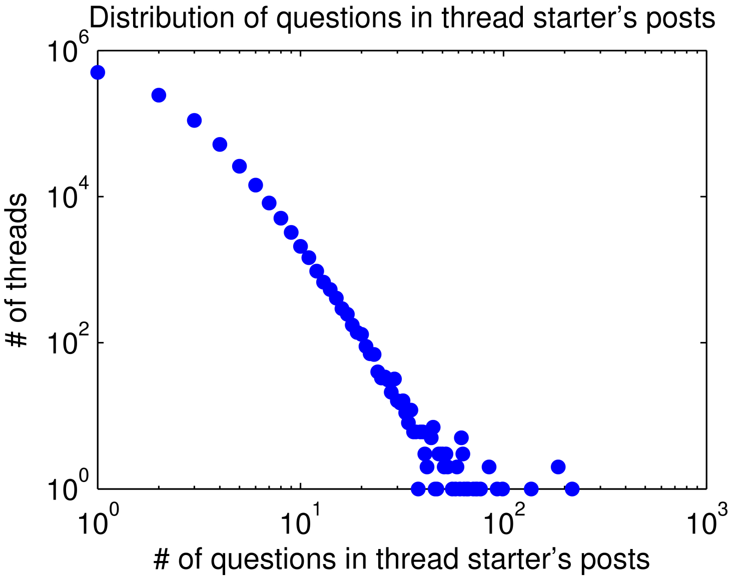


Figure 5.6: Question level distribution of thread starters' activity.

rule-based classification algorithm Ripper to perform the question detection task. We used GIZA++ [154] to train the IBM model 1. We used GibbsLDA++ [166] to conduct LDA training and inference.

Metrics: For the evaluation of the task, we adopted several well-known metrics that evaluate different aspects of the performance of the proposed method, including Precision at Rank R (P@R), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Kullback-Leibler divergence (KL-divergence). Reciprocal rank is an accepted measure in question answering evaluation. It favors hits that are ranked higher, however, gives appropriate weights to lower ranked hits [228].

Parameter Tuning: There are several parameters need to be determined in our experiments. We used 20 queries from the TST_LABEL, and employed MAP to tune the parameters. Optimal parameters are as follows: $\alpha = 0.25$, $\beta = 0.1$, $K = 200$, $\lambda = 2,000$, $\delta = 0.2$, and $\gamma = 0.7$.

Table 5.3: The first row shows the source words. Top 10 words that are most semantically related to the source word are presented according to IBM translation model 1 and LDA. All the words are lowercased and stemmed.

Words	**shore**		**park**		**condo**		**beach**	
Rank	**IBM 1**	**LDA**	**IBM 1**	**LDA**	**IBM 1**	**LDA**	**IBM 1**	**LDA**
1	shore	shore	park	park	condo	condo	beach	beach
2	beach	groceri	drive	hotel	beach	south	resort	slope
3	snorkel	thrift	car	stai	area	north	what	jet
4	island	supermarket	how	time	unit	shore	hotel	snowboard
5	kauai	store	area	area	island	pacif	water	beaver
6	condo	nappi	where	recommend	maui	windward	walk	huski
7	area	tesco	walk	beach	rent	seaport	area	steamboat
8	water	soriana	time	nation	owner	alabama	room	jetski
9	boat	drugstor	ride	tour	shore	opposit	snorkel	powder
10	ocean	mega	hotel	central	rental	manor	restaur	hotel

Experiment on Word Translation

To answer RQ1, we used the proposed method to build the parallel corpus, and the constructed parallel corpus contains $2,629,533$ question-question pairs. Table 5.3 shows the top 10 words that are most semantically related to the given words employing IBM model 1 and LDA.

Various semantic relationships between words were discovered using IBM model 1. For example, when a user is asking a question about shore, snorkel is related because snorkeling is a popular activity in shore, and condo is also related because the user also needs to rent a condo for living. Walton is a beach name in Florida's Emerald Coast near Pensacola and Destin. Its full name is Fort Walton Beach. Atlanta is also related to Walton because the nearest Airport of Walton provides frequent flights to Atlanta. Recall that the proposed method considers that questions in a thread could translate to each other, leading to capturing the semantic relationships of words from semantically related questions. In other words, it

Table 5.4: Comparison on Labeled Questions (a larger metric value means a better performance).

Metrics	**LDA**	**QL**	**TR**	**TRLM**	**TopicTRLM**
$P@R$	0.2411	0.3370	0.4135	0.4555	**0.5140**
MAP	0.3684	0.4089	0.4629	0.5029	**0.5885**
MRR	0.5103	0.5277	0.5311	0.5317	**0.5710**

characterizes relations in related events that happen in related questions. We could find that LDA captures different relations, and the reason is that LDA describes co-occurrence relations because it considers words in a question. For example, people ask questions like "Is there any grocery store at Orange Beach?", and LDA is capable of capturing this kind of word relations between grocery and beach in a sentence. Thus, we believe both approaches capture different semantic aspects between words.

Experiment on Labeled Question

We conducted an experiment on TST_LABEL to answer RQ2. We employed the word-to-word translation probabilities learnt from the parallel question-question corpus in TR, TRLM, TopicTRLM. The experimental results on metrics P@R, MAP, and MRR are shown in Table 5.4. All the results are statistically significant according to the sign test compared with the previous method.

Table 5.4 shows that LDA performs the worst. Because LDA is a coarse-grained representation to measure the relatedness between questions, it is not able to capture accurate meaning of each question. TR has better question suggestion performance compared with QL. This finding is consistent with the previous work [87, 86]. The reason is that the translation model has the potential to bridge the lexical chasm between related questions. It also confirms the effectiveness of the proposed method to build parallel corpus of related questions from forum thread. TRLM has better performance than

TR because TR set the probability of self-translation to 1. This introduces inconsistent probability estimates and makes the model unstable. The proposed TopicTRLM outperforms other approaches in all metrics. This confirms the effectiveness of TopicTRLM in the question suggestion task. The advantage of TopicTRLM compared with other approaches is that it fuses the latent semantic meanings of questions with lexical similarities, and this fusion promises to benefit from both the bag-of-words representation and topic model representation.

Experiment on Topics' Joint Probability Distribution

To answer RQ3, we conducted experiments on TST_UNLABEL to evaluate topic level performances of the proposed method. For each queried question q, we consider its first subsequent question q' posted by the thread starter in the actual thread as its relevant result. For all the $10,000$ queried questions and their relevant results, we used the trained LDA model to infer the most probable topic. We aggregated the counts of topic transitions in the actual threads as ground truth and applied maximum likelihood estimation approach to calculate topics' joint probability using Eq. 5.8:

$$p(topic(q), topic(q')) = p(topic(q')|topic(q)) \times p(topic(q)). \quad (5.8)$$

We used a 200×200 ($K = 200$) matrix to represent ground truth topics' joint probability distributions. In addition, for each queried question, we employed different approaches to retrieve results and considered the first result as its suggested question. We measured the difference between two probability distributions using the Kullback-Leibler divergence. Experimental results in Table 5.5 confirm the effectiveness of the proposed TopicTRLM.

Table 5.5: Comparison on difference between ground truth and methods' topics' joint probability distribution (a smaller KL-divergence value means a better performance).

Methods	**Kullback-Leibler Divergence**
LDA	0.1127
QL	0.1067
TR	0.0955
TRLM	0.0911
TopicTRLM	**0.0906**

5.3.2 Experiments in Community-based Q&A Services

Experimental Setup

Methods: To evaluate the performance of the proposed methods, we compared the proposed algorithms with alternative approaches. Specifically, we compared our methods topic-enhanced translation-based language model (TopicTRLM) and topic-enhanced translation-based language model with answer ensemble (TopicTRLM-A) with LDA [27], query likelihood language model using Dirichlet smoothing (QL) [250] and translation-based language model (TRLM) [244].

Data set: We used Yahoo! Answers dataset from Yahoo! Webscope program [245]. The dataset includes $4,483,032$ questions and their answers. More specifically, we utilized the resolved questions under two of the top-level categories at Yahoo! Answers, namely "travel" and "computers & internet". We randomly sampled 100 questions from each category as test questions. The remaining questions and their corresponding best answers in each category were used as question repository, as well as training set for learning word translation probabilities and building LDA model. We used the unigram language model to represent questions, and applied IBM model 1 to learn unigram to unigram translation probabilities. We used Porter Stemmer [171] to stem question words. We adopted the stop word list used by SMART system [33], but 5W1H words were removed from the stop word list. For each model, the top 20 retrieval results were kept. We used pooling [136] to put results from different models for one query together for annotation, and all models were used in the pooling process. If a returned result was considered as semantically related to the queried question, it was labeled with "relevant"; otherwise, it was labeled with "irrelevant". Two assessors were involved in the initial labeling process. If two assessors had different opinions on a decision, a third assessor was asked to make a final decision. The kappa statistics between two assessors was 0.78. We used GIZA++ [154] to train the IBM model 1. We used GibbsLDA++ [166] to conduct LDA training and inference.

Metrics: For the evaluation of the task, we adopted several well-known metrics that evaluate different aspects of the performance of the proposed method, including Precision at Rank R (P@R), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and Bpref. Bpref is proposed by Buckley et al. [32], and is the score function of the number of non-relevant candidates.

Parameter Tuning: There are several parameters need to be determined in our experiments. We used 20 queries from each category, and employed MAP to tune the parameters. Optimal parameters are as follows: $\alpha = 0.25$, $\beta = 0.1$, $K = 200$, $\lambda = 2,000$, $\delta = 0.2$, $\gamma = 0.7$, $\epsilon = 0.7$, $\eta = 0.2$, $\theta = 0.6$ and $\mu = 0.2$.

Experiment on Yahoo! Answers Dataset

Table 5.6 demonstrates the results of different models on category "computers and internet", and Table 5.7 shows the results on category "travel". All the results are statistically significant according to the sign test compared with the previous method.

From Table 5.6 and Table 5.7, we can find that TopicTRLM-A achieves the best performance on different metrics on two categories. The reason is TopicTRLM-A combines contributions from both questions and their answers through utilizing lexical and latent semantic relatedness, thus getting the best performance. Top-

Table 5.6: Performance of different models on category “computers & internet” (a larger metric value means a better performance).

Methods	**MAP**	**Bpref**	**MRR**	**P@R**
LDA	0.2397	0.136	0.2767	0.1594
QL	0.346	0.2261	0.416	0.2594
TRLM	0.3532	0.2368	0.4271	0.2777
TopicTRLM	0.4235	0.2755	0.5559	0.3197
TopicTRLM-A	**0.6228**	**0.4673**	**0.7745**	**0.5467**

Table 5.7: Performance of different models on category ”travel” (a larger metric value means a better performance).

Methods	**MAP**	**Bpref**	**MRR**	**P@R**
LDA	0.1345	0.0612	0.1616	0.0675
QL	0.316	0.1902	0.388	0.2048
TRLM	0.3222	0.2034	0.3923	0.2234
TopicTRLM	0.3615	0.244	0.4406	0.2644
TopicTRLM-A	**0.467**	**0.3167**	**0.5963**	**0.387**

icTRLM performs better than methods which utilizes lexical only or latent topic information only by fusing both the lexical and latent semantic knowledge. TRLM performs better than QL, which is consistent with previous research [244]. LDA performs the worst since it is too coarse grained.

We also look into concrete results for test questions. Table 5.8 and Table 5.9 show results for two test questions from the category “computers and internet”. Table 5.10 shows results for one test question from the category “travel”. Let’s take the question in Table 5.10 for example, the queried question is “Why can people only use the air phones when flying on commercial airlines, i.e. no cell phones etc.?” Thus, the underlying information need of the user is to know why cell phone could not be used in commercial airlines. The first result of TopicTRLM-A model is “Why are you supposed to keep cell phone off during flight in commercial airlines?” We can find the first result is semantically equivalent to the test question, thus, the best answer of the first result should answer the user's information need accurately. The second result of TopicTRLM-A model is "Why don't cell phones from the ground at or near airports cause interference in the communications of aircraft?" This question is quite related to the test question since it also discusses the interference of cell phones to the communications of aircraft, and it also belongs to the topic of "interference of aircraft". The third result is "Cell phones and pagers really dangerous to avionics?" This question would open the asker's mind that not only cell phones, but also pagers maybe dangerous to aircraft systems, more specifically, to avionics. We can find that TopicTRLM-A could not only find questions that are semantically equivalent to the queried question, but also find questions that are semantically related to the queries question. Thus, TopicTRLM-A could satisfy users' information needs more thoroughly. Table 9 and Table 10 show similar findings.

We also test the sensitivity of parameter ϵ in TopicTRLM-A. A larger ϵ means we reply more on lexical score, and a smaller ϵ means we reply more on latent semantic knowledge. Figure 5.7 shows the MAP of employing different ϵ on category "computers and internet".

Figure 5.8 shows the Bpref of different approaches.

Figure 5.9 shows the different models on MRR.

Figure 5.10 shows the P@R for different approaches.

TopicTRLM-A performs the best when ϵ is between 0.5 and 0.7 on different metrics. The relatively wide optimal parameter ranges indicates that only by fusing both lexical and latent semantic knowledge together, the model could achieve the best performance. Figure 5.7, Fig. 5.8, Fig. 5.9, and Fig. 5.10 also demonstrate that TopicTRLM-A is sensitive to the parameter ϵ, but the parameter ϵ is still feasible to tune. The optimal parameter range of ϵ is similar on the other category "travel".

Table 5.8: The results for “Hi, I lost my Yahoo password? How can I get my old password back without any changing with my email?” of category “computers & internet”.

Methods	**Results**
LDA	1. How can I send my MSN password to my other account if my MSN password is lost?
	2. I lost my administrator password and I only have a guest as a user how can I get my password or another one?
	3. My other Yahoo email password is stolen by someone, **how can I report it and get it back as soon as possible?**
QL	1. I keep having a problem with my password. It keeps changing my password or not letting me sign on?
	2. I need a program that can help me figure out a password without actually changing the password, or altering it?
	3. My minor daughter’s Yahoo name and password were changed by someone along with them changing email. What do I?
TRLM	1. I need a program that can help me figure out a password without actually changing the password, or altering it?
	2. My minor daughter’s Yahoo name and password were changed by someone along with them changing email. What do I?
	3. I’m on Myspace and I changed my password but I forgot my password and my email password. What do I do?
TopicTRLM	1. I lost my administrator password and I only have a guest as a user how can I get my password or another one?
	2. I forgot my security question, how will I get my lost Yahoo password back for my other id?
	3. I have lost my Yahoo password. I don’t remember any of the information I fed into the sign u form?
TopicTRLM-A	**1. I have lost my Yahoo password. I don’t remember any of the information I fed into the sign u form?**
	2. I forgot my security question, how will I get my lost Yahoo password back for my other id?
	3. If I forget my Yahoo password, is there any way to get it back or does Yahoo have to send me a new one?

Table 5.9: The results for “I want to just know how to use Outlook Express to send mail, and if possible to sync with Yahoo or MSN?” of category “computers & internet”.

Methods	**Results**
LDA	**1. How can I do to synchronize my Yahoo or MSN email account with Microsoft Outlook to send and receive messages?**
	2. How can you set up to read your sent messages on MSN Hotmail, like Yahoo?
	3. What’s my Yahoo incoming mail (POP3/IMAP)?!! At Yahoo mail & outgoing mail (SMTP) required 4 email notifier?
QL	**1. Can I sync Outlook Express inbox with Yahoo inbox?**
	2. Is there a way to sync Yahoo calendar to Blackberry 7100t without having Outlook?
	3. How do I sync contacts from Ipaq with Outlook? Most of them are not getting synched?
TRLM	**1. Can I sync Outlook Express inbox with Yahoo inbox?**
	2. I have several contacts in my Outlook that are not showing up on my Treo when I sync. How do I fix???
	3. Is there a way to sync Yahoo calendar to Blackberry 7100t without having Outlook?
TopicTRLM	1. What’s my Yahoo incoming mail (POP3/IMAP)?!! At Yahoo mail & outgoing mail (SMTP) required 4 email notifier?
	2. How do I synchronize my Yahoo mail with Outlook Express i.e. I wish to use Outlook Express to check my Y! mail.
	3. How can I do to synchronize my Yahoo or MSN email account with Microsoft Outlook to send and receive messages?
TopicTRLM-A	**1. I want to use Yahoo mail in my Outlook Express. Please tell me POP3 and SMTP address of Yahoo?**
	2. Is it possible to configure my Yahoo id in Outlook Express 6?
	3. Sir, please tell me I am all Yahoo mail converlet in Outlook please tell me how I do?

Table 5.10: The results for “Why can people only use the air phones when flying on commercial airlines, i.e. no cell phones etc.?” of category “travel”.

Methods	**Results**
LDA	**1, Why are you supposed to keep cell phone off during flight in commercial airlines?**
	2, I will be flying from CA to FL. Any tips on how I can get over my fear of flying?
	3, I need the contact number of emirates airlines here in Philippines?
QL	**1, Cell phones and pagers really dangerous to avionics?**
	2, Do cell phones work on cruise ships? T-mobile?
	3, Cell phones in Singapore, Bali or Kuala Lumpur?
TRLM	**1, Why don’t cell phones from the ground at or near airports cause interference in the communications of aircraft?**
	2, Should I bring my Vertu phones in my carry-on luggage or send through? I have 3 of them?
	3, Cell phones and pagers really dangerous to avionics?
TopicTRLM	**1, Cell phones and pagers really dangerous to avionics?**
	2, Why don’t cell phones from the ground at or near airports cause interference in the communications of aircraft?
	3, Do cell phones work on cruise ships? T-mobile?
TopicTRLM-A	**1, Why are you supposed to keep cell phone off during flight in commercial airlines?**
	2, Why don’t cell phones from the ground at or near airports cause interference in the communications of aircraft?
	3, Cell phones and pagers really dangerous to avionics?

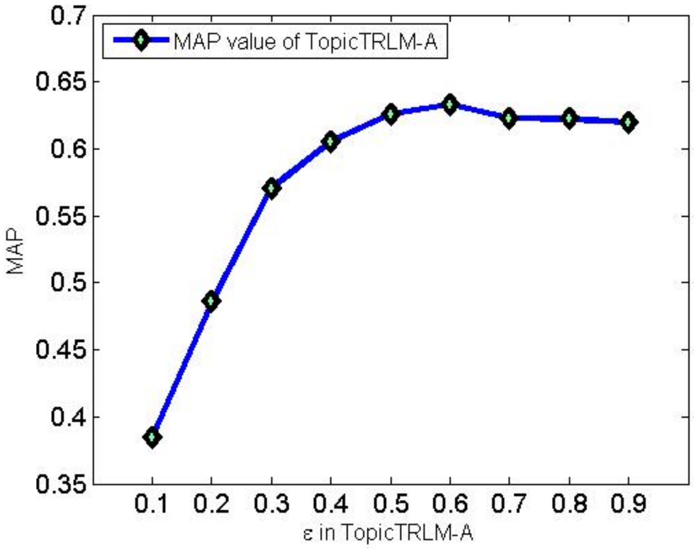


Figure 5.7: The effect of parameter ϵ on the MAP of question suggestion.

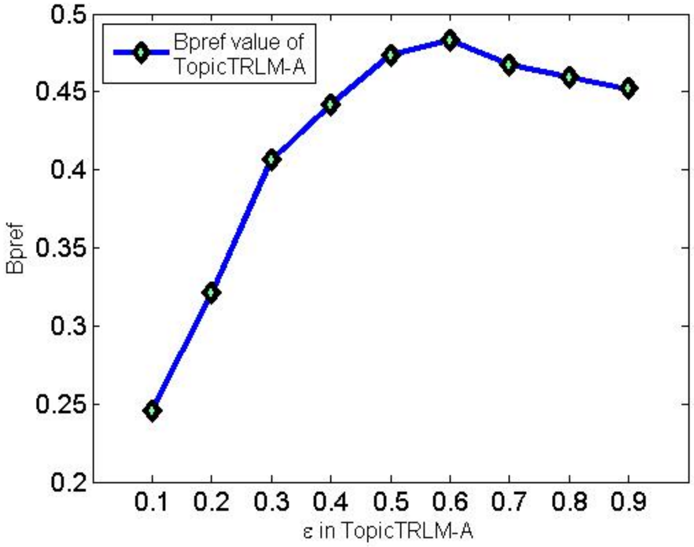


Figure 5.8: The effect of parameter ϵ on the Bpref of question suggestion.

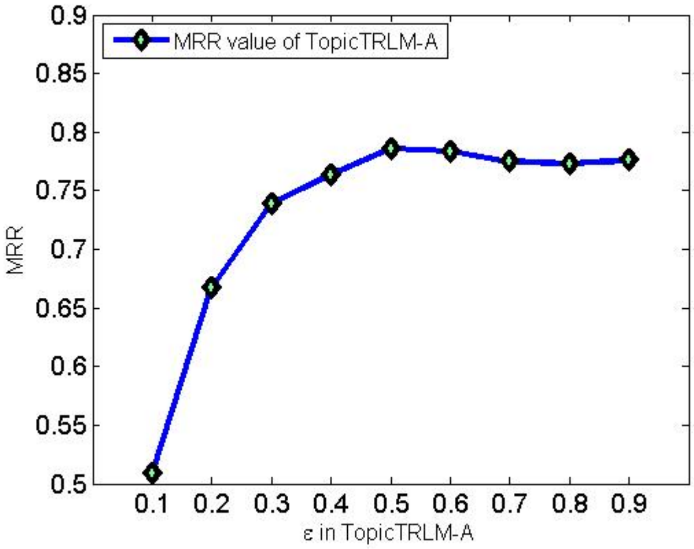


Figure 5.9: The effect of parameter ϵ on the MRR of question suggestion.

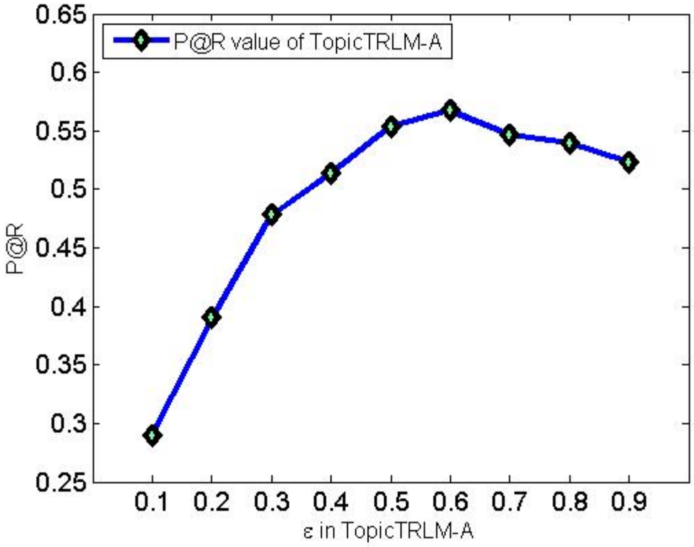


Figure 5.10: The effect of parameter ϵ on the P@R of question suggestion.

5.4 Summary

In this chapter we address the issue of question suggestion in social media. Given a queried question, we are to suggest questions that are semantically related to the queried question and can explore different aspects of a topic tailored to users' information needs. We tackle the problem on two types of the most representative social media systems: online forums and community-based Q&A services. In online forums, we propose an effective method to build the parallel corpus of related questions from forum threads, and we propose TopicTRLM, which fuses lexical knowledge with latent semantic knowledge to measure the relatedness between questions. In community-based Q&A services, we also propose an effective method to build the parallel corpus of related questions, and we propose TopicTRLM-A, which incorporates answer information into question to measure the semantic relatedness more thoroughly. Extensive experiments indicate that our method to build parallel corpus is effective and the TopicTRLM and TopicTRLM-A methods outperform other approaches.

□ **End of chapter.**

Chapter 6

Item Modeling via Data-Driven Approach

6.1 Problem and Motivation

In this chapter we focus on item modeling via data-driven approach. Specifically, we focus on question subjectivity identification for questions in social media systems. Automatic Subjective Question Answering (ASQA), which aims at answering users' subjective questions using summaries of multiple opinions, becomes increasingly important. One challenge of ASQA is that expected answers for subjective questions may not readily exist in the Web. The rising and popularity of Community Question Answering (CQA) sites, which provide platforms for people to post and answer questions, provides an alternative to ASQA. One important task of ASQA is question subjectivity identification, which identifies whether a user is asking a subjective question. Unfortunately, there has been little labeled training data available for this task. In this paper, we propose an approach to collect training data automatically by utilizing social signals in CQA sites without involving any manual labeling. Experimental results show that our data-driven approach achieves 9.37% relative improvement over the supervised approach using manually labeled data, and achieves 5.15% relative gain over a state-of-the-art semi-supervised approach. In addition, we propose several heuristic features for question subjectivity identification. By adding these features, we achieve 11.23% relative improvement over word n-gram feature under the same experimental setting.

6.2 Question Subjectivity Identification

We treat question subjectivity identification as a classification task. Subjective questions are considered as positive instances, and objective questions are considered as negative instances. In this section, we propose several social signals for collecting training data, and propose several heuristic features for the QSI task.

6.2.1 Social Signal Investigation

Like (L): in CQA sites, users *like* an answer if they find the answer is useful. Even the best answer of a question has been chosen, users could *like* other answers as well as the best answer. The intuition of the *like* signal is as follows: answers posted to a subjective question are opinions. Due to different tastes of the large community of users, not only the best answer, but also other answers may receive *likes* from users. Thus, if the best answer receives similar number of *likes* with other answers, it is very likely that the question is subjective. If a question is objective, the majority of users would *like* an answer which explains universal truth or common knowledge in the most detailed manner. Thus, the best answer would receive extremely high *likes* than other answers. Equation (6.1) presents the criteria of selecting positive training data:

$$L(Q_{best\_answer}) \leq \frac{\sum L(Q_{answer})}{AN(Q)}, \tag{6.1}$$

where $L(\cdot)$ is the number of people *like* this answer, Q_{best_answer} is the best answer of a question Q, Q_{answer} is an answer of a question Q, and $AN(\cdot)$ is the number of answers of a question. Equation (6.2) presents the criteria of selecting negative training data:

$$L(Q_{best\_answer}) \geq \alpha \times MAX(L(Q_{other\_answer})), \tag{6.2}$$

where α is a parameter, Q_{other_answer} is an answer except the best answer of a question Q, and $MAX(\cdot)$ is the maximum function. *Like* signal is commonly found in CQA sites, such as *rate* in Yahoo! Answers, *support* in Baidu Knows, and *like* in AnswerBag[1].

Vote (V): users could vote for the best answer in CQA sites. An answer that receives the most *votes* is chosen as the best answer. The intuition of *vote* signal is as follows: the percentage of *votes* of the best answer of an objective question should be high, since it is relatively easy to identify which answer contains the most thorough universal truth or common knowledge. However, users may *vote* for different answers of a subjective question since they may support different opinions, resulting in a relatively low percentage of *votes* on the best answer. Equation (6.3) shows the criteria of selecting positive training data:

$$V(Q_{best\_answer}) \leq \beta, \tag{6.3}$$

where $V(\cdot)$ is the percentage of votes of an answer, and β is a parameter. Equation (6.4) shows the criteria of selecting negative training data:

$$V(Q_{best\_answer}) \geq \gamma, \tag{6.4}$$

where γ is a parameter. There is *vote* signal in many popular CQA sites, such as Yahoo! Answers, Baidu Knows, and Quora[2].

Source (S): to increase the chance of an answer to be selected as the best answer, users often provide *sources* of their answers. A *source* of an answer is a reference to authoritative resources. The intuition of *source* signal is that *source* is only available for an objective question that has a fact answer. For an subjective question, users just post their opinions without referencing authorities. In our

[1] http://answerbag.com

[2] http://quora.com

Table 6.1: Social signals investigated to collect training data and their descriptions.

Name	Description
Like	Social signal that captures users' tastes on an answer.
Vote	Social signal that reflects users' judgments on an answer.
Source	Social signal that measures users' confidence on authoritativeness of an answer.
Poll and Survey	Social signal that indicates users' intent of a question.
Answer Number	Social signal that implies users' willingness to answer a question.

approach, we collect questions with *source* as negative training data. *Source* signal exists in many popular CQA sites such as Yahoo! Answers, Baidu Knows, and AnswerBag.

Poll and Survey (PS): since a large number of community users are brought together in CQA sites, users often post *poll and survey* questions. The intuition of *poll and survey* signal is that the user intent of a *poll and survey* question is to seek opinions on a certain topic. Thus, a *poll and survey* question is very likely to be a subjective question. In addition, CQA sites often have mechanisms to enable users to post *poll and survey* questions. For example, Yahoo! Answers has a dedicated category named *poll and survey*. In our approach, we collect *poll and survey* questions as positive training data.

Answer Number (AN): the number of posted answers to each question in CQA sites varies a lot. The intuition of *answer number* signal is as follows: users may post opinions to a subjective question even they notice there are other answers for the question. Thus, the number of answers of a subjective question may be large. However, users may not post answers to an objective question that has already received other answers since an expected answer is usually fixed. Thus, a large *answer number* may indicate subjectivity of a question, but a small *answer number* may be due to many reasons, such as objectivity and small page views. Equation (6.5) presents the criteria

Table 6.2: Social signals investigated and the type of training data that could be collected.

Name	Training Data
Like	Positive and Negative.
Vote	Positive and Negative.
Source	Negative.
Poll and Survey	Positive.
Answer Number	Positive.

of collecting positive training data.

$$AN(Q) \geq \theta, \tag{6.5}$$

where $AN(\cdot)$ is the number of answers of a question, and θ is a parameter. Table 6.1 summarizes all social signals that are investigated in this study and their descriptions.

Table 6.2 summarizes social signals investigated and the type of training data that could be collected.

6.2.2 Feature Investigation

Word (word): word feature is shown to be effective in many question answering applications. We also study this feature in this paper. Specifically, each word is represented with its term frequency (tf) value.

Word n-gram (ngram): we utilize word n-gram feature. Previous supervised [110] and small scale semi-supervised [109] approaches on QSI observed that the performance gain of word n-gram compared with word feature was not significant, but we conjecture that it may be due to the sparsity of their small amount of labeled training data. We investigate whether word n-gram would have significant gain if we have a large amount of training data. Specifically, each word n-gram is represented with its tf value.

Besides basic features, we also study several light-weight heuristic features in this paper. These heuristic features could be computed efficiently, leading to the scalability of proposed approach.

Question length (qlength): information needs of subjective questions are complex, and users often use descriptions [234] to explain their questions, leading to larger question length. We investigate whether question length would help QSI. We divide question length into 10 buckets, and the corresponding bucket number is used as a feature.

Request word (rword): we observe that in CQA sites, users use some particular words to explicitly indicate their request for seeking opinions. We refer to these words as *request words*. Specifically, 9 words are manually selected, i.e. “should”, “might”, “anyone”, “can”, “shall”, “may”, “would”, “could”, and “please”. The total number of request words is used as a feature.

Subjectivity clue (sclue): we investigate whether external lexicons would help QSI. Specifically, in this study, we utilize subjectivity clues from the work of Wilson et al. [239], which contain a lexicon of over 8000 subjectivity clues. Subjectivity clues are manually compiled word lists that may be used to express opinions, i.e., they have subjective usages.

Punctuation density (pdensity): punctuation density is measured according to the density of punctuation marks in questions. Equation (6.6) presents the formulation of calculating punctuation density for a question:

$$PDensity(Q) = \frac{\text{\# punctuation marks}}{\text{\# punctuation marks} + \text{\# words}}. \tag{6.6}$$

Grammatical modifier (gmodifier): inspired by opinion mining research of using grammatical modifiers on judging users’ positive and negative opinions, we investigate the effectiveness of using grammatical modifier as a feature. Specifically, adjective and adverb are considered as grammatical modifiers.

Entity (entity): the expected answer for an objective question is fact or common knowledge, leading to less relationships among entities compared with a complex subjective question. Thus, we conjecture that the number of entities varies between subjective and objective questions. Specifically, we use noun as the surrogate of entity in our study.

6.3 Experimental Evaluation

6.3.1 Experimental Setting

Comparison methods: the baseline approach of question subjectivity identification was supervised learning using labeled training data. In addition, we compared with the state-of-the-art approach CoCQA proposed by Li et al. in [109]. CoCQA was a co-training approach that exploits the association between the questions and contributed answers.

Dataset: the raw data that was used to collect training data using social signals was from Yahoo! Answers, and there was $4,375,429$ questions with associated answers and social signals. They were relatively popular questions according to user behaviors, and were actively indexed with high priority in our system. They could be considered as reusable and valuable resources. In Yahoo! Answers data, *rate* function was used as the *like* signal, *vote* function was used as the *vote* signal, *source* field in the best answer was used as the *source* signal, the category *poll and survey* was used as the *poll and survey* signal, and *number of answers* was used as the *answer number* signal. Social signals investigated in this study are quite general, and other CQA sites could be leveraged to collect training data as well. The ground truth data set we used was adapted from Li et al. [109]. They created the data set using Amazon's Mechanical Turk service[3]. As suggested in [173, 246], we used a sampling method to deal with the imbalance problem in their data set, i.e. to keep all objective questions and randomly sample the same number of subjective questions. We obtained 687 questions in total, and we

[3] http://www.mturk.com

referred it as T. We also employed sampling method when using social signals to collect training data. The same with Li et al. [109], we reported the average results of 5-fold cross validation on T for supervised learning and CoCQA. Unlabeled data for CoCQA was from Liu et al. [119]. The results of our approach on T were also reported for comparison. It is worthwhile to point out that our approach did not use any manually labeled data. To tune the parameters for different social signals, 20% of questions in T were randomly selected. This data set was used as the development set, and referred to as D.

Classification method: we employed Naive Bayes with add-one smoothing classification method [48] in our experiments. Aikawa et al. [4] found Naive Bayes was more effective than Support Vector Machines [73] in classifying subjective and objective questions. In addition, the training process of Naive Bayes was able to be parallelized using MapReduce framework [50].

Metric: precision on subjective questions was used as the evaluation metric in our experiments. The reason was as follows: a user's satisfaction would be increased if he/she receives an answer that summarizes people's opinions for a subjective question, but his/her satisfaction would not be decreased if he/she receives an answer the same with existing CQA sites that are not equipped with subjective question identification component. A user's satisfaction would be decreased if he/she receives a summarized answer that repeats the fact for an objective question. Thus, precision on subjective questions was the appropriate metric.

Parameter tuning: we performed grid search using different parameter values over D. We ran grid search from 1.0 to 2.5 for α in *like* signal, from 0.1 to 1.0 for β and γ in *vote* signal alternatively, and from 10 to 30 for θ in *answer number* signal. The optimal setting was as follows: $\alpha = 2.0$, $\beta = 0.2$, $\gamma = 0.5$ and $\theta = 20$.

Table 6.3: Performance of supervised, CoCQA, and combinations of social signals with the word n-gram feature. Value in parenthesis means relative performance gain compared with supervised approach.

Method	**Precision**
Supervised	0.6596
CoCQA	0.6861 (+4.20%)
L + V + PS + AN + S	0.6626 (+0.45%)
L	0.5714 (−13.37%)
V + PS + AN + S	0.6981 (+5.84%)
PS + AN + S	0.6915 (+4.84%)
V + PS + AN	**0.7214** (+**9.37**%)
V + AN	0.7201 (+9.17%)
AN + S	0.7038 (+6.70%)

6.3.2 Effectiveness of Social Signals

We employed different combinations of social signals to automatically collect positive and negative training data, and used the trained classifier to identify subjective questions. Table 6.3 presents the results using word n-gram feature. Specifically, we employed unigram and bigram for word n-gram. By employing co-training over questions and associated answers, CoCQA utilizes some amount of unlabeled data, and achieves better results than supervised approach. However, similar with [109], we found CoCQA achieved optimal performance after adding $3,000$ questions. It means CoCQA could only utilize a small amount of unlabeled data considering the large volume of CQA archives.

In Table 6.3, it is promising to observe that collecting training data using social signals $V + PS + AN$ achieves the best results. It improves 9.37% and 5.15% relatively over supervised and CoCQA respectively. The results indicate the effectiveness of collecting training data using well-designed social signals for QSI. Selecting training data using $V+AN$ and $AN+S$ achieve the second and third best performance. Both combinations perform better than supervised and

CoCQA. In addition, social signals of V, AN, S could be found in almost all CQA sites. Due to the page limit, we report results of several combinations of social signals. Other combinations achieve comparable performances. Collecting training data using $like$ signal does not perform well. We look into the training data, and find that some objective questions are considered as subjective because their best answers receive fewer *likes* than other answers. Considering the fact that many best answers are chosen by the asker, we conjecture that this phenomenon may be due to the complex of best answer selection criteria in CQA sites. Previous work also found socio-emotional factor affected a lot in the best answer selection [94]. We leave the detailed study of how users choose their best answers to our future work.

Table 6.4 reports the results of different approaches using word and word n-gram feature. In line with our intuition, all approaches achieve better performance using word n-gram feature compared with word feature. More interestingly, we find that combinations of social signals, $V + PS + AN$, $V + AN$ and $AN + S$ achieve on average 12.27% relative gain of employing word n-gram over word. But supervised approach only achieves 3.39%, and CoCQA achieves 6.66% relative gain of using word n-gram over word. We conjecture the reason is as follows: supervised approach only utilizes manually labeled training data, resulting in the sparsity of employing word n-gram. CoCQA uses several thousand unlabeled data, and tackles the sparsity problem to some extent. Training data collected according to social signals is quite large compared with previous approaches, and data sparsity problem is better solved.

Table 6.5 reports the performance of three best performing combinations of social signals with varying amount of training data using word n-gram. With the increase of training data, performances of three approaches all improve accordingly. This finding is encouraging because in practical, we may integrate training data from several CQA sites with the same social signal.

Table 6.4: Performance of different approaches using word and word n-gram. Value in parenthesis means relative performance gain of word n-gram compared with word.

Method/Feature	**Word**	**Word n-gram**
Supervised	0.6380	0.6596 (+3.39%)
CoCQA	0.6432	0.6861 (+6.66%)
V + PS + AN	0.6707	0.7214 (+7.56%)
V + AN	0.6265	0.7201 (+14.94%)
AN + S	0.6157	0.7038 (+14.31%)

Table 6.5: Performance of three best performing combinations of social signals with varying training data.

	20%	**40%**	**90%**	**100%**
V + AN	0.6549	0.7004	0.7188	0.7201
AN + S	0.6550	0.6696	0.6842	0.7038
V + PS + AN	0.6640	0.6846	0.7037	0.7214

6.3.3 Effectiveness of Heuristic Features

Previously, we discussed results of utilizing social signals to automatically collect training data. In this section, we study the effectiveness of heuristic features. To allow others to repeat our results, experiments investigating heuristic features were conducted on the data set T, which contains 687 questions adapted from Li et al. [109].

Question length: Fig. 6.1 shows the proportion of subjective questions (denoted as Prop-Sub) with respect to questions' lengths. We rank the questions according to their lengths in ascending order, and equally partition them into 10 groups. Figures 6.3, 6.4, 6.5, and 6.6 apply similar methods to show Prop-Sub with respect to the corresponding features. Interestingly, we find the proportion of subjective questions increases as the question length increases. To find out the reason, we look into the data, and observe that when a user asks an objective question, he/she just expresses his/her information

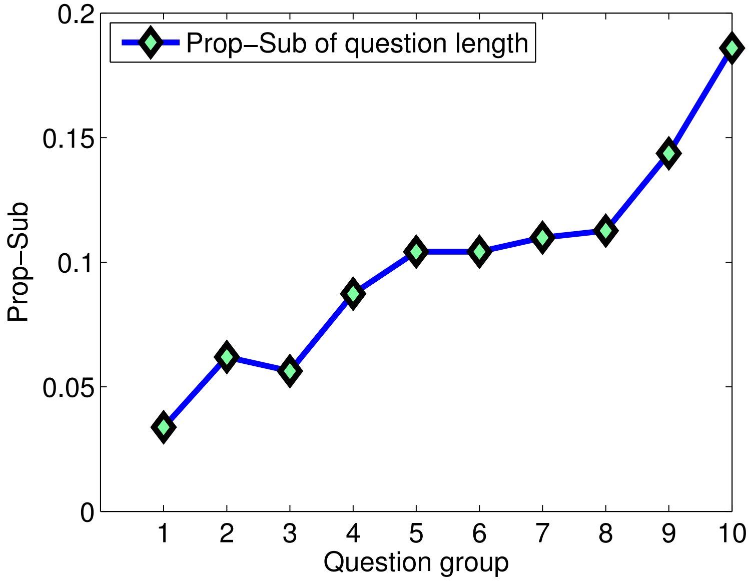


Figure 6.1: The question length feature.

needs precisely, e.g., “Which player has won the fa cup twice with 2 different teams?” However, when a user asks a subjective question, he/she also shares his/her personal opinion together with the question, e.g., “Has anyone read “Empire” by Orson Scott Card? This is scary. I especially liked the “Afterword” by him. It’s amazing how close you can feel today to it coming true.”

Request word: Fig. 6.2 demonstrates the percentage of subjective questions (denoted as Perc-Sub) with respect to the number of request word. Group 1 contains questions that don’t have any request word, group 2 contains questions having 1 request word, group 3 contains 2 request words, group 4 contains 3 request words, and group 5 contains at least 4 request words. Perc-Sub measures the percentage of subjective questions among all questions in each group. Quite surprisingly, we find Perc-Sub increases as the number of request words increases. After checking some sample questions, we conclude the reason is that when users ask subjective questions, they also add complicated background or detailed opinions, making the question quite long. To attract potential answerers, users add

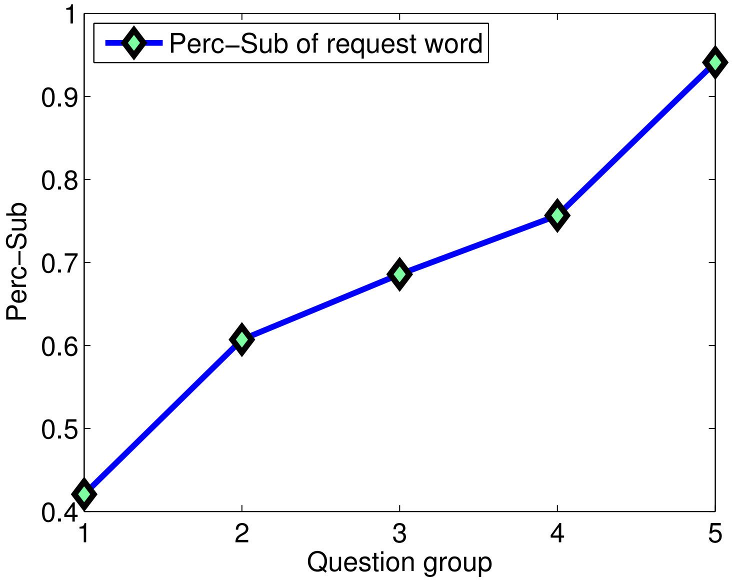


Figure 6.2: The request word feature.

these request words.

Subjective clue: in Fig. 6.3, we can see a clear trend that the more subjective clues, the larger proportion of subjective questions. This is an interesting finding that although subjective clues used in our experiments are from other documents, such as news, they still help distinguish between subjective and objective questions to some extent.

Punctuation density: in Fig. 6.4, we observe that the higher punctuation density, the higher proportion of subjective questions. In other words, the punctuation mark density of subjective questions is higher than that of objective questions. After examining some examples, we find that users use short sentence segments when sharing their experiences in subjective questions. In addition, we conjecture that short sentence segments help better express users' feelings and opinions in asking subjective questions.

Grammatical modifier: in Fig. 6.5, we find the proportion of subjective questions is positively correlated with the number of grammatical modifiers. The reason comes from the observation that gram-

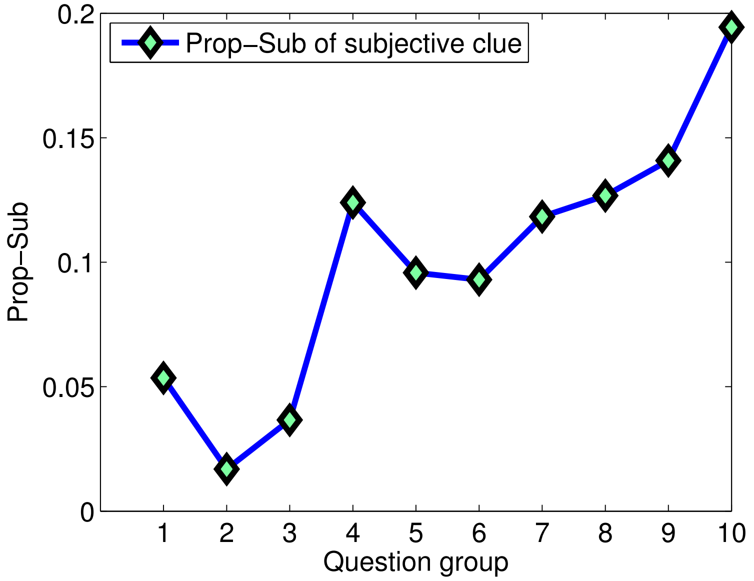


Figure 6.3: The subjective clue feature.

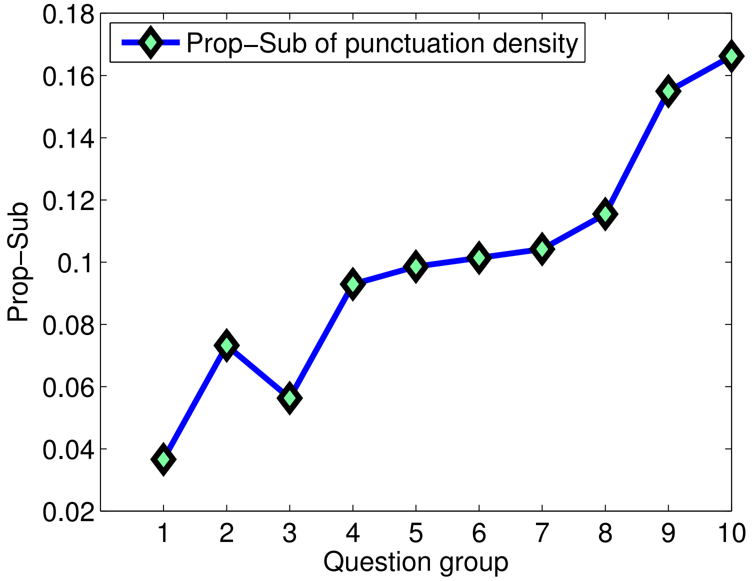


Figure 6.4: The punctuation density feature.

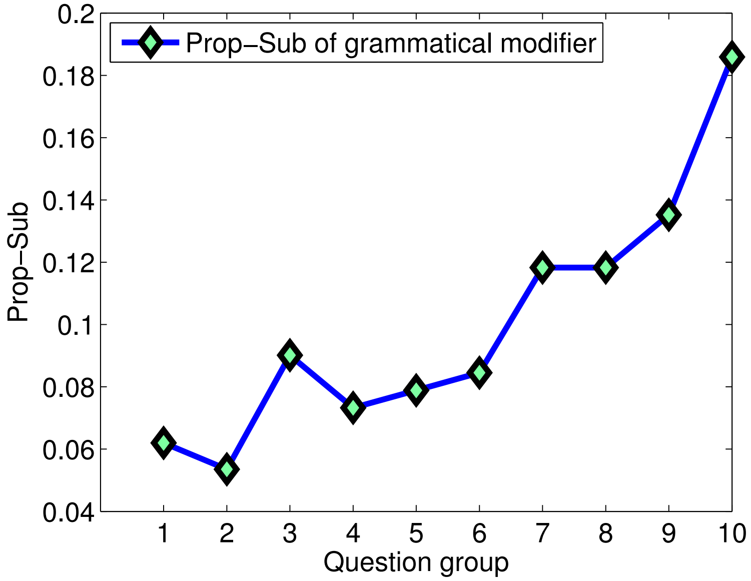


Figure 6.5: The grammatical modifier feature.

matical modifiers are commonly used to describe users' feelings, experiences, and opinions in subjective questions. Thus, the more grammatical modifiers used, the larger proportion of subjective questions.

Entity: it is interesting to observe from Fig. 6.6 that the proportion of subjective question increases as the number of entities increases. After investigating some samples, we find that information needs of objective questions involve fewer entities compared with subjective questions. The reason is that subjective questions involve more descriptions, which also contain entities.

Table 6.6 shows results of employing heuristic features and word n-gram. We observe that adding any heuristic feature to word n-gram would improve precision to some extent, and employing only heuristic features performs even better than word n-gram. Combining heuristic features and word n-gram achieves 11.23% relative performance gain over employing word n-gram.

Table 6.7 shows examples of questions wrongly classified using n-gram, but correctly classified with the incorporation of heuristic

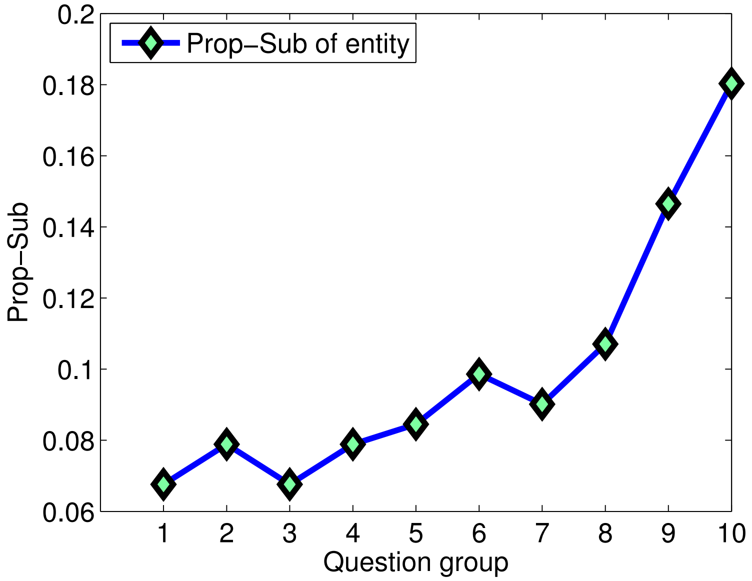


Figure 6.6: The entity feature.

Table 6.6: Performance of heuristic features.

Precision	**ngram**	**ngram** **+ qlength**	**ngram** **+ rword**	**ngram** **+ sclue**
	0.6596	0.6896	0.6834	0.6799
ngram **+ pdensity**	**ngram** **+ gmodifier**	**ngram** **+ entity**	**heuristic** **features**	**ngram** **+ heuristic**
0.7000	0.6950	0.6801	0.6995	$\mathbf{0.7337}(+\mathbf{11.23}\%)$

Table 6.7: Examples of questions wrongly classified using n-gram, but correctly classified with the incorporation of heuristic features.

Examples
Who is Mugabe?
When and how did Tom Thompson die? He is one of the group of seven.
Was Roy Orbison blind?
How is an echocardiogram done?
Fluon Elastomer material's detail?
What does BCS stand for in college football?

features. These results demonstrate the effectiveness of proposed heuristic features.

6.4 Summary

In this chapter, we present a data-driven approach for utilizing social signals in CQA sites. We demonstrate our approach for one particular important task of automatically identifying question subjectivity, showing that our approach is able to leverage social interactions in CQA portals. Despite the inherent difficulties of question subjectivity identification for real user questions, we have demonstrated that our approach can significantly improve prediction performance than the supervised approach [110] and a state-of-the-art semi-supervised approach [109]. We also study various heuristic features for QSI, and experimental results confirm the effectiveness of proposed features.

□ **End of chapter.**

Chapter 7

Conclusion

7.1 Summary

This thesis establishes automatic and scalable models to help social media users find their information needs more effectively. These models are proposed based on the two key entities in social media systems: user and item. This thesis develops a framework to combine the user information and item information with the following two purposes: 1) modeling users' interests with respect to their behavior, and recommending items or users they may be interested in; and 2) understanding items' characteristics, and grouping items that are semantically related for better addressing users' information needs.

For the first purpose, we present two user-based models with applications. Firstly, to overcome the data sparsity problem and non-flexibility problem confronted by traditional collaborative filtering algorithms, we propose a factor analysis approach, referred to as *TagRec*, by utilizing both users' rating information and tagging information based on probabilistic matrix factorization. Secondly, to provide users with an automatic and effective way to discover other users with common interests in social tagging systems, we propose the *User Recommendation (UserRec)* framework for user interest modeling and interest-based user recommendation, aiming to boost information sharing among users with similar interests. Specifically, we propose a tag-graph based community detection method to model the users' personal interests, which are further represented by discrete topic distributions. The similarity values between users' topic distributions are measured by Kullback-Leibler divergence (KL-divergence), and the similarity values are further used to perform interest-based user recommendation.

For the second purpose, we present two item-oriented models with applications. Firstly, we propose a new functionality *Question Suggestion*, which targets at suggesting questions that are semantically related to a queried question, in social media systems with Q&A functionalities. Existing bag-of-words approaches suffer from the shortcoming that they could not bridge the lexical chasm between semantically related questions. Therefore, we present a new framework to suggest questions, and propose the *Topic-enhanced Translation-based Language Model (TopicTRLM)* which fuses both the lexical and latent semantic knowledge. Moreover, to incorporate the answer information into the model to make the model more complete, we also propose the *Topic-enhanced Translation-based Language Model with Answer Ensemble (TopicTRLM-A)*. Secondly, to improve the performance of question subjectivity identification in community-based Q&A services with the constrain that little labeled training data are available, we propose an approach to collect training data automatically by utilizing social signals in community-based Q&A sites without involving any manual labeling. In addition, we propose several heuristic features for question subjectivity identification.

7.2 Future Work

Although a substantial number of promising achievements on techniques and its applications have been presented in this thesis, there are still numerous open issues that need to be further explored in future work.

Firstly, the proposed TagRec approach uses the explicit relations directly, such as users' rating information and tagging information; the approach also considers each user and each item equally, ignoring the fact that there may be some hidden structures among all the users and all the items. In the future, we will investigate whether it is possible to first mine these explicit relations to infer some implicit relations, and then use the inferred implicit relations and the original explicit relations together to improve the recommendation quality.

Secondly, we would like to extend proposed UserRec approach and develop a more robust framework that can handle the *tag ambiguity* problem. Moreover, we plan to investigate how information, such as URLs and tags, is propagated in the social tagging systems.

Thirdly, because we want to assist users in exploring different aspects of the topic that he/she is interested in by offering question suggestion service, it is worthwhile to investigate how to measure and how to diversify the suggested questions. Moreover, as question suggestion improves systems' understanding of users' latent intent, query suggestion for long queries might also benefit from question suggestion, which is also a future direction to investigate.

Fourthly, we plan to explore more sophisticated features such as semantic analysis using natural language processing techniques. We will investigate characteristics of subjective questions, and study whether we could find popular semantic patterns for subjective questions. We will also apply our data-driven framework to tasks such as sentiment analysis in community-based Q&A services and to similar social media platforms.

□ **End of chapter.**

Appendix A

List of Publications

A.1 Conference Publications

1. **Tom Chao Zhou**, Xiance Si, Edward Y. Chang, Irwin King and Michael R. Lyu. A Data-Driven Approach to Question Subjectivity Identification in Community Question Answering. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI-12)*, pp 164-170, Toronto, Ontario, Canada, July 22 - 26, 2012.

2. **Tom Chao Zhou**, Michael R. Lyu and Irwin King. A Classification-based Approach to Question Routing in Community Question Answering. In *Proceedings of the 21st International Conference Companion on World Wide Web*, pp 783-790, Lyon, France, April 16 - 20, 2012.

3. **Tom Chao Zhou**, Chin-Yew Lin, Irwin King, Michael R. Lyu, Young-In Song and Yunbo Cao. Learning to Suggest Questions in Online Forums. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*, pp 1298-1303, San Francisco, California, USA, August 7 - 11, 2011.

4. Zibin Zheng, **Tom Chao Zhou**, Michael R. Lyu, and Irwin King. FTCloud: A Ranking-based Framework for Fault Tolerant Cloud Applications. In *Proceedings of the 21st IEEE*

International Symposium on Software Reliability Engineering (ISSRE 2010), pp 398-407, San Jose CA, USA, November 1-4, 2010.

5. **Tom Chao Zhou**, Hao Ma, Michael R. Lyu, Irwin King. UserRec: A User Recommendation Framework in Social Tagging Systems. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*, pp 1486-1491, Atlanta, Georgia, USA, July 11 - 15, 2010.

6. **Tom Chao Zhou**, Irwin King. Automobile, Car and BMW: Horizontal and Hierarchical Approach in Social Tagging Systems. In *Proceedings of the 2nd Workshop on Social Web Search and Mining (SWSM 2009), in conjunction with CIKM 2009*, pp 25-32, Hong Kong, November 2 - 6, 2009.

7. **Tom Chao Zhou**, Hao Ma, Irwin King, Michael R. Lyu. TagRec: Leveraging Tagging Wisdom for Recommendation. In *Proceedings of the 15th IEEE International Conference on Computational Science and Engineering (CSE-09)*, pp 194199, Vancouver, Canada, 29-31 August, 2009.

A.2 Journal Publications

1. Zibin Zheng, **Tom Chao Zhou**, Michael R. Lyu, and Irwin King. Component Ranking for Fault-Tolerant Cloud Applications, *IEEE Transactions on Service Computing (TSC)*, 2011.

2. Hao Ma, **Tom Chao Zhou**, Michael R. Lyu and Irwin King. Improving Recommender Systems by Incorporating Social Contextual Information, *ACM Transactions on Information Systems (TOIS)*, Volume 29, Issue 2, 2011.

A.3 Under Review

1. **Tom Chao Zhou**, Michael R. Lyu and Irwin King. Learning to Suggest Questions in Social Media. Submitted to *Journal of the American Society for Information Science and Technology (JASIST)*.

□ **End of chapter.**

Bibliography

[1] L.A. Adamic, J. Zhang, E. Bakshy, and M.S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th International Conference on World Wide Web*, pages 665–674. ACM, 2008.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] E. Agichtein, S. Lawrence, and L. Gravano. Learning search engine specific query transformations for question answering. In *Proceedings of the 10th International Conference on World Wide Web*, pages 169–178. ACM, 2001.

[4] Naoyoshi Aikawa, Tetsuya Sakai, and Hayato Yamana. Community qa question classification: Is the asker looking for subjective answers or not? *IPSJ Online Transactions*, pages 160–168, 2011.

[5] M.R. Anderberg. Cluster analysis for applications. Technical report, DTIC Document, 1973.

[6] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *Proceeding of the 17th International Conference on World Wide Web*, pages 199–208. ACM, 2008.

[7] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro. User modeling and recommendation techniques for personalized electronic program guides. *Personalized Digital Television*, pages 3–26, 2004.

[8] J.S. Armstrong. *Principles of forecasting: a handbook for researchers and practitioners*, volume 30. Springer, 2001.

[9] H. Avancini and U. Straccia. User recommendation for collaborative and personalised digital archives. *International Journal of Web Based Communities*, 1(2):163–175, 2005.

[10] R. Badi, S. Bae, J.M. Moore, K. Meintanis, A. Zacchi, H. Hsieh, F. Shipman, and C.C. Marshall. Recognizing user interest and document value from reading and organizing activities in document triage. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, pages 218–225. ACM, 2006.

[11] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York., 1999.

[12] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R.J. Mooney. Model-based overlapping clustering. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 532–537. ACM, 2005.

[13] P. Bedi, H. Kaur, and S. Marwaha. Trust based recommender system for semantic web. In *Proceedings of the 2007 International Joint Conferences on Artificial Intelligence*, pages 2677–2682, 2007.

[14] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. *Learning Theory*, pages 624–638, 2004.

[15] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1):209–239, 2004.

[16] N.J. Belkin and W.B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[17] R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 95–104. ACM, 2007.

[18] R.M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[19] R.M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, pages 43–52. IEEE, 2007.

[20] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–199. ACM, 2000.

[21] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 222–229. ACM, 1999.

[22] P. Berkhin. A survey of clustering data mining techniques. *Grouping Multidimensional Data*, pages 25–71, 2006.

[23] D. Bernhard and I. Gurevych. Combining lexical semantic resources with question & answer archives for translation-based answer finding. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 728–736. Association for Computational Linguistics, 2009.

[24] J. Bian, Y. Liu, E. Agichtein, and H. Zha. Finding the right facts in the crowd: factoid question answering over social media. In *Proceedings of the 17th International Conference on World Wide Web*, pages 467–476. ACM, 2008.

[25] N. Bila, J. Cao, R. Dinoff, T.K. Ho, R. Hull, B. Kumar, and P. Santos. Mobile user profile acquisition through network observables and explicit user queries. In *Proceedings of 9th International Conference on Mobile Data Management*, pages 98–107. IEEE, 2008.

[26] D. Billsus, M.J. Pazzani, et al. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 54, page 48, 1998.

[27] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[28] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100. ACM, 1998.

[29] P. Bonhard, M.A. Sasse, and C. Harries. The devil you know knows best: how online recommendations can benefit from social networking. In *Proceedings of the 21st British HCI*

Group Annual Conference on People and Computers, pages 77–86. British Computer Society, 2007.

[30] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[31] P.F. Brown, J. Cocke, S.A.D. Pietra, V.J.D. Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[32] C. Buckley and E.M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR*, pages 25–32, 2004.

[33] Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. New retrieval approaches using smart: Trec 4. In *Proceedings of TREC*, pages 25–48, 1995.

[34] R.D. Burke, K.J. Hammond, V. Kulyukin, S.L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57, 1997.

[35] J. Canny. Collaborative filtering with privacy. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 45–57. IEEE, 2002.

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

[37] X. Cao, G. Cong, B. Cui, C.S. Jensen, and Q. Yuan. Approaches to exploring category information for question retrieval in community question-answer archives. *ACM Transactions on Information Systems (TOIS)*, 30(2):7, 2012.

[38] Y. Cao, H. Duan, C.Y. Lin, and Y. Yu. Re-ranking question search results by clustering questions. *Journal of the American Society for Information Science and Technology*, 62(6):1177–1187, 2011.

[39] Y. Cao, H. Duan, C.Y. Lin, Y. Yu, and H.W. Hon. Recommending questions using the mdl-based tree cut model. In *Proceedings of the 17th International Conference on World Wide Web*, pages 81–90. ACM, 2008.

[40] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. MIT Press Cambridge, MA, 2006.

[41] H. Cheng, Y. Zhou, X. Huang, and J.X. Yu. Clustering large attributed information networks: an efficient incremental computing approach. *Data Mining and Knowledge Discovery*, pages 1–28, 2012.

[42] Y.H. Chien and E.I. George. A bayesian model for collaborative filtering. In *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, 1999.

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

[44] G. Cong, L. Wang, C.Y. Lin, Y.I. Song, and Y. Sun. Finding question-answer pairs from online forums. In *Proceedings of the 31st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 467–474. ACM, 2008.

[45] F. Crestani, M. Lalmas, C.J. Van Rijsbergen, and I. Campbell. Is this document relevant? a survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998.

[46] W.B. Croft and D.J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4):285–295, 1979.

[47] W.B. Croft and J. Lafferty. *Language modeling for information retrieval*, volume 13. Springer, 2003.

[48] W.B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. 2010.

[49] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 293–296. ACM, 2010.

[50] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[51] J. Delgado and N. Ishii. Memory-based weighted majority prediction. In *ACM SIGIR'99 Workshop on Recommender Systems*. Citeseer, 1999.

[52] D. Demner-Fushman and J. Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103, 2007.

[53] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[54] H. Duan, Y. Cao, C.Y. Lin, and Y. Yu. Searching questions by identifying question topic and question focus. In *Proceedings of ACL*, pages 156–164, 2008.

[55] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern classification. *New York: John Wiley, Section*, 10:l, 2001.

[56] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A.A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.

[57] N. Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992.

[58] S. Funk. Netflix update: Try this at home. Technical report, sifter.org/simon/journal/ 20061211.html, 2006.

[59] R. Gazan. Social q&a. *Journal of the American Society for Information Science and Technology*, 62(12):2301–2312, 2011.

[60] T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM)*, page 625C628. IEEE, 2005.

[61] L. Getoor and M. Sahami. Using probabilistic relational models for collaborative filtering. In *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*. Citeseer, 1999.

[62] J. Golbeck. Generating predictive movie recommendations from trust in social networks. *Trust Management*, pages 93–104, 2006.

[63] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[64] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

[65] G.H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[66] T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[67] V.N. Gudivada, V.V. Raghavan, W.I. Grosky, and R. Kasanagottu. Information retrieval on the world wide web. *IEEE Internet Computing*, 1(5):58–68, 1997.

[68] I. Guy, M. Jacovi, A. Perer, I. Ronen, and E. Uziel. Same places, same things, same people?: mining user similarity on social media. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 41–50. ACM, 2010.

[69] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, pages 8–12, 2009.

[70] S. Harabagiu, D. Moldovan, M. Pasca, M. Surdeanu, R. Mihalcea, R. Girju, V. Rus, F. Lacatusu, P. Morarescu, and R. Bunescu. Answering complex, list and context questions with lcc's question-answering server. In *Proceedings of the TExt Retrieval Conference for Question Answering (TREC 10)*, 2001.

[71] J.A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[72] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: data mining, inference, and prediction, 2001.

[73] M.A. Hearst, ST Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and Their Applications, IEEE*, 13(4):18–28, 1998.

[74] G. Heinrich. Parameter estimation for text analysis. Technical report, Fraunhofer IGD, 2005.

[75] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237. ACM, 1999.

[76] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of the International Conference on Web Search and Web Data Mining*, pages 195–206. ACM, 2008.

[77] D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. *Research and Advanced Technology for Digital Libraries*, pages 515–515, 1998.

[78] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[79] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[80] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115, 2004.

[81] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In *International Joint Conference on Artificial Intelligence*, volume 16, pages 688–693, 1999.

[82] J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to automata theory, languages, and computation*, volume 2. Addison-wesley Reading, MA, 1979.

[83] W. Hosmer David and L. Stanley. Applied logistic regression. *Wiley-Interscience Publication*, 2000.

[84] S. Huston and W.B. Croft. Evaluating verbose query processing techniques. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 291–298. ACM, 2010.

[85] S.Y. Hwang and L.S. Chen. Using trust for collaborative filtering in ecommerce. In *Proceedings of the 11th International Conference on Electronic Commerce*, pages 240–248. ACM, 2009.

[86] J. Jeon, W.B. Croft, and J.H. Lee. Finding semantically similar questions based on their answers. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 617–618. ACM, 2005.

[87] J. Jeon, W.B. Croft, and J.H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 84–90. ACM, 2005.

[88] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 76–83. ACM, 2005.

[89] R. Jin, J.Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 337–344. ACM, 2004.

[90] R. Jin, A.G. Hauptmann, and C.X. Zhai. Language model for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–48. ACM, 2002.

[91] R. Jin, L. Si, and C. Zhai. A study of mixture models for collaborative filtering. *Information Retrieval*, 9(3):357–382, 2006.

[92] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[93] K.S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[94] S. Kim, J.S. Oh, and S. Oh. Best-answer selection criteria in a social q&a site from the user-oriented relevance perspective. *Proceedings of ASIST*, 44(1):1–15, 2007.

[95] S. Kim and S. Oh. Users' relevance criteria for evaluating answers in a social q&a site. *Journal of the American Society for Information Science and Technology*, 60(4):716–727, 2009.

[96] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.

[97] A. Kohrs and B. Merialdo. Clustering for collaborative filtering applications. In *Proceedings of CIMCA*, pages 199–204, 1999.

[98] J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[99] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434. ACM, 2008.

[100] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[101] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM, 2002.

[102] Vinod Krishnan, Pradeep Kumar Narayanashetty, Mukesh Nathan, Richard T. Davies, and Joseph A. Konstan. Who predicts better?: Results from an online study comparing humans and an online recommender system. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, 2008.

[103] J. Krumm, N. Davies, and C. Narayanaswami. User-generated content. *IEEE Pervasive Computing*, 7(4):10–11, 2008.

[104] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Recommendation systems: A probabilistic analysis. In *Proceedings the 39th Annual Symposium on Foundations of Computer Science*, pages 664–673. IEEE, 1998.

[105] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 194–201. ACM, 2004.

[106] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. *Language Modeling for Information Retrieval*, 13:1–10, 2003.

[107] V. Lavrenko and W.B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 120–127. ACM, 2001.

[108] W.C. Lee and E.A. Fox. Experimental comparison of schemes for interpreting boolean queries. 1988.

[109] B. Li, Y. Liu, and E. Agichtein. Cocqa: co-training over questions and answers with an application to predicting question subjectivity orientation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 937–946. Association for Computational Linguistics, 2008.

[110] B. Li, Y. Liu, A. Ram, E.V. Garcia, and E. Agichtein. Exploring question subjectivity prediction in community qa. In *Proceedings of SIGIR*, pages 735–736, 2008.

[111] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.Y. Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2008.

[112] X. Li, L. Guo, and Y.E. Zhao. Tag-based social interest discovery. In *Proceedings of the 17th International Conference on World Wide Web*, pages 675–684. ACM, 2008.

[113] G.L. Lilien, P. Kotler, and K.S. Moorthy. *Marketing models*. Prentice Hall, 1992.

[114] J. Lin and B. Katz. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861, 2006.

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

[116] N.N. Liu, X. Meng, C. Liu, and Q. Yang. Wisdom of the better few: cold start recommendation via representative based rating elicitation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 37–44. ACM, 2011.

[117] N.N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–90. ACM, 2008.

[118] X. Liu and W.B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193. ACM, 2004.

[119] Y. Liu, J. Bian, and E. Agichtein. Predicting information seeker satisfaction in community question answering. In *Proceedings of SIGIR*, pages 483–490, 2008.

[120] J. Lou, Y. Fang, K.H. Lim, and J.Z. Peng. Contributing high quantity and quality knowledge to online q&a communities. *Journal of the American Society for Information Science and Technology*, 2013.

[121] J. Lou, Y. Fang, K.H. Lim, and Z.J. Peng. Knowledge contribution in online question and answering communities: Effects of groups membership. In *Proceedings of 2012 International Conference on Information Systems*, 2012.

[122] J. Lou, K. Lim, Y. Fang, and Z. Peng. Drivers of knowledge contribution quality and quantity in online question and answering communities. In *Proceedings of the 15th Pacific Conference on Information Systems*, 2011.

[123] J. Lou, K.H. Lim, Y.L. Fang, and Z.Y. Peng. Drivers of knowledge contribution quality and quantity in online question and answering communities. In *Proceedings of PACIS*, 2011.

[124] K. Lu, J. Lafferty, S. Zhu, and Y. Gong. Large-scale collaborative prediction using a nonparametric random effects model. In *ICML*, 2009.

[125] H.P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[126] H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[127] H. Ma. *Learning to Recommend.* PhD thesis, The Chinese University of Hong Kong, 2009.

[128] H. Ma, I. King, and M.R. Lyu. Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):29, 2011.

[129] H. Ma, D. Zhou, C. Liu, M.R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, pages 287–296. ACM, 2011.

[130] H. Ma, T.C. Zhou, M.R. Lyu, and I. King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, 29(2):9, 2011.

[131] Hao Ma, Irwin King, and Michael R. Lyu. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–46, 2007.

[132] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 931–940, 2008.

[133] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, 2010.

[134] Z. Ma, O.R.L. Sheng, and G. Pant. Evaluation of ontology-based user interests modeling. In *Proceedings of the 4th Workshop on e-Business*, pages 512–518, 2005.

[135] P. Maes et al. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–40, 1994.

[136] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[137] B. Marlin. Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[138] B. Marlin and R.S. Zemel. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 73. ACM, 2004.

[139] M.E. Maron and J.L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

[140] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508, 2004.

[141] P. Massa and B. Bhattacharjee. Using trust in recommender systems: an experimental analysis. *Trust Management*, pages 221–235, 2004.

[142] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, volume 752, pages 41–48. Citeseer, 1998.

[143] P. Melville, R.J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the National Conference on Artificial Intelligence*, pages 187–192. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[144] D.R.H. Miller, T. Leek, and R.M. Schwartz. A hidden markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221. ACM, 1999.

[145] G.A. Miller et al. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[146] T.M. Mitchell et al. Machine learning, 1997.

[147] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214. ACM, 1998.

[148] C. Mooers. From a point of view of mathematical etc. techniques. *Towards Information Retrieval*, 17:528, 1961.

[149] G. Muresan and D.J. Harper. Topic modeling for mediated access to very large document collections. *Journal of the American Society for Information Science and Technology*, 55(10):892–910, 2004.

[150] BPS Murthi and S. Sarkar. The role of the management sciences in research on personalization. *Management Science*, 49(10):1344–1362, 2003.

[151] A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 395–403, 1998.

[152] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131+, 2004.

[153] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 86–93. ACM, 2000.

[154] F.J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[155] J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174. ACM, 2005.

[156] J. O'Donovan and B. Smyth. Is trust robust?: an analysis of trust-based recommendation. In *Proceedings of the 11th International Conference on Intelligent User Interfaces*, pages 101–108. ACM, 2006.

[157] J. O'Donovan and B. Smyth. Mining trust values from recommendation errors. *International Journal on Artificial Intelligence Tools*, 15(06):945–962, 2006.

[158] B. Ofoghi, J. Yearwood, and L. Ma. The impact of frame semantic annotation levels, frame-alignment techniques, and fusion methods on factoid answer processing. *Journal of the American Society for Information Science and Technology*, 60(2):247–263, 2009.

[159] W. Pan, E.W. Xiang, and Q. Yang. Transfer learning in collaborative filtering with uncertain ratings. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[160] M. Papagelis, D. Plexousakis, and T. Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. *Trust Management*, pages 125–140, 2005.

[161] K. Papineni. Why inverse document frequency? In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[162] D. Pavlov, E. Manavoglu, C.L. Giles, and D.M. Pennock. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems*, 19(6):40–47, 2004.

[163] D.Y. Pavlov and D.M. Pennock. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. *Advances in Neural Information Processing Systems*, 15:1441–1448, 2002.

[164] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine learning*, 27(3):313–331, 1997.

[165] M.J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5):393–408, 1999.

[166] X.H. Phan, L.M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, pages 91–100. ACM, 2008.

[167] G. Pitsilis and L. Marshall. Trust as a key to improving recommendation systems. *Trust Management*, pages 421–432, 2005.

[168] G. Pitsilis and L.F. Marshall. *A model of trust derivation from evidence for use in recommendation systems*. University of Newcastle upon Tyne, Computing Science, 2004.

[169] J. Pomerantz. A linguistic analysis of question taxonomies. *Journal of the American Society for Information Science and Technology*, 56(7):715–728, 2005.

[170] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281. ACM, 1998.

[171] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[172] M.J.D. Powell. *Approximation theory and methods*. Cambridge university press, 1981.

[173] F. Provost. Machine learning from imbalanced data sets. In *AAAI Workshop on Imbalanced Data Sets*, 2000.

[174] B. Qu, G. Cong, C. Li, A. Sun, and H. Chen. An evaluation of classification models for question topic categorization. *Journal of the American Society for Information Science and Technology*, 2012.

[175] J.R. Quinlan. Bagging, boosting, and c4.5. In *Proceedings of the National Conference on Artificial Intelligence*, pages 725–730, 1996.

[176] D.R. Raban. Self-presentation and the value of information in q&a websites. *Journal of the American Society for Information Science and Technology*, 60(12):2465–2473, 2009.

[177] D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering on the web. *Journal of the American Society for Information Science and Technology*, 56(6):571–583, 2005.

[178] D.R. Radev, K. Libner, and W. Fan. Getting answers to natural language questions on the web. *Journal of the American Society for Information Science and Technology*, 53(5):359–364, 2002.

[179] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[180] J.D.M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings*

of the 22th International Conference on Machine Learning (ICML), 2005.

[181] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994.

[182] E. Rich. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

[183] S. Riezler, Y. Liu, and A. Vasserman. Translating queries into snippets for improved query expansion. In *Proceedings of COLING*, pages 737–744, 2008.

[184] S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[185] S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, pages 42–49. ACM, 2004.

[186] S.E. Robertson. The probability ranking principle in ir. *Journal of Documentation*, 33(4):294–304, 1977.

[187] S.E. Robertson and K.S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 2007.

[188] S.E. Robertson, C.J. van Rijsbergen, and M.F. Porter. Probabilistic models of indexing and searching. In *Proceedings of*

the 3rd Annual ACM Conference on Research and Development in Information Retrieval, pages 35–56. Butterworth & Co., 1980.

[189] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995.

[190] H. Rosenbaum and P. Shachaf. A structuration approach to online communities of practice: The case of q&a communities. *Journal of the American Society for Information Science and Technology*, 61(9):1933–1944, 2010.

[191] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. 2005.

[192] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine learning*, pages 880–887, 2008.

[193] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, 20, 2008.

[194] G. Salton. The smart retrieval system - experiments in automatic document processing. 1971.

[195] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[196] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[197] G. Salton, E.A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[198] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[199] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, DTIC Document, 2000.

[200] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, pages 285–295. ACM, 2001.

[201] E. Savia, K. Puolamaki, J. Sinkkonen, and S. Kaski. Two-way latent grouping model for user preference prediction. *arXiv preprint arXiv:1207.1414*, 2012.

[202] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. Tagging, communities, vocabulary, evolution. In *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work*, pages 181–190, 2006.

[203] C. Shah and V. Kitzie. Social q&a and virtual referencecomparing apples and oranges with the help of experts and users. *Journal of the American Society for Information Science and Technology*, 2012.

[204] G. Shani, R.I. Brafman, and D. Heckerman. An mdp-based recommender system. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 453–460. Morgan Kaufmann Publishers Inc., 2002.

[205] J. Shapiro, V.G. Voiskunskii, and V.I. Frants. *Automated information retrieval: theory and methods*. Academic Press Professional, Inc., 1997.

[206] U. Shardanand and P. Maes. Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co., 1995.

[207] R. Sheldon et al. *A first course in probability*. Pearson Education India, 2002.

[208] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.

[209] X. Si, E.Y. Chang, Z. Gyöngyi, and M. Sun. Confucius and its intelligent disciples: Integrating social with search. volume 3, pages 1505–1516, 2010.

[210] Xiance Si, Zhiyuan Liu, Peng Li, Qixia Jiang, and Maosong Sun. Content-based and graph-based tag suggestion. In *Proceedings of the ECML PKDD Discovery Challenge*, pages 243–260, 2009.

[211] V. Sindhwani and S.S. Keerthi. Large scale semi-supervised linear svms. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 477–484. ACM, 2006.

[212] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing & Management*, 32(5):619–633, 1996.

[213] P. Singla and M. Richardson. Yes, there is a correlation - from social networks to personal behavior on the web. In

Proceedings of the 17th International Conference on World Wide Web, pages 655–664. ACM, 2008.

[214] A. Smola and R. Kondor. Kernels and regularization on graphs. *Learning Theory and Kernel Machines*, pages 144–158, 2003.

[215] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *Proceedings of HLT-NAACL*, pages 191–206, 2004.

[216] K. Sparck Jones, S. Walker, and S.E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779–808, 2000.

[217] K. Sparck Jones, S. Walker, and S.E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840, 2000.

[218] N. Srebro, T. Jaakkola, et al. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 720–727, 2003.

[219] N. Srebro, J.D.M. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems*, 17(5):1329–1336, 2005.

[220] Fabian M. Suchanek, Milan Vojnovic, and Dinan Gunawardena. Social tags: meaning and suggestions. In *CIKM*, pages 223–232, 2008.

[221] M. Szomszor, H. Alani, I. Cantador, K. OHara, and N. Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. *ISWC*, pages 632–648, 2008.

[222] T. Tao, X. Wang, Q. Mei, and C.X. Zhai. Language model information retrieval with document expansion. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 407–414. Association for Computational Linguistics, 2006.

[223] M. Taube and H. Wooster. *Information storage and retrieval: theory, systems and devices*. Number 10. Columbia University Press, 1958.

[224] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[225] L. Ungar and D.P. Foster. A formal statistical approach to collaborative filtering. *CONALD98*, 1998.

[226] L.H. Ungar and D.P. Foster. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems*, 1998.

[227] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.

[228] E. Voorhees and D.M. Tice. The trec-8 question answering track evaluation. In *Proceedings of The Eighth Text REtrieval Conference (TREC-8)s*, 1999.

[229] F.E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[230] J. Wang, A. de Vries, and M. Reinders. A user-item relevance model for log-based collaborative filtering. *ECIR*, pages 37–48, 2006.

[231] J. Wang, A.P. De Vries, and M.J.T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 501–508. ACM New York, NY, USA, 2006.

[232] J. Wang, A.P. de Vries, and M.J.T. Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 26(3):16, 2008.

[233] K. Wang, Z. Ming, and T.S. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 187–194. ACM, 2009.

[234] Kai Wang, Zhao-Yan Ming, Xia Hu, and Tat-Seng Chua. Segmentation of multi-sentence questions: Towards effective question retrieval in cqa services. In *Proceedings of SIGIR*, pages 387–394, 2010.

[235] X. Wei and W.B. Croft. Lda-based document models for adhoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185. ACM, 2006.

[236] J. Weng, C. Miao, and A. Goh. Improving collaborative filtering with trust-based metrics. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, pages 1860–1864. ACM, 2006.

[237] R. Wetzker, C. Zimmermann, and C. Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Pro-*

ceedings of Mining Social Data Workshop on ECAI, pages 26–30, 2008.

[238] R.W. White, P. Bailey, and L. Chen. Predicting user interests from contextual information. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 363–370. ACM, 2009.

[239] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of EMNLP*, pages 347–354, 2005.

[240] S.K.M. Wong, W. Ziarko, and P.C.N. Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25. ACM, 1985.

[241] Y.H. Wu, Y.C. Chen, and A.L.P. Chen. Enabling personalized recommendation on the web based on user interests and behaviors. In *Proceedings of Eleventh International Workshop on Research Issues in Data Engineering*, pages 17–24. IEEE, 2001.

[242] J. Xu and W.B. Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11. ACM, 1996.

[243] G.R. Xue, C. Lin, Q. Yang, W.S. Xi, H.J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 114–121, 2005.

[244] X. Xue, J. Jeon, and W.B. Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 475–482. ACM, 2008.

[245] Yahoo! Yahoo! webscope dataset, ydata-yanswers-all-questions-v1_0, http://research.yahoo.com/academic_relations.

[246] L. Yang, S. Bao, Q. Lin, X. Wu, D. Han, Z. Su, and Y. Yu. Analyzing and predicting not-answered questions in community-based question answering services. In *Proceedings of AAAI*, pages 1273–1278, 2011.

[247] K. Yu, S. Zhu, J. Lafferty, and Y. Gong. Fast nonparametric matrix factorization for large-scale collaborative filtering. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 211–218. ACM, 2009.

[248] H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian extension to the language model for ad hoc information retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–9. ACM, 2003.

[249] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342. ACM, 2001.

[250] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.

[251] C.X. Zhai. Statistical language models for information retrieval: a critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, 2008.

[252] C.X. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–56. ACM, 2002.

[253] S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. *SIAM*, 2006.

[254] Zibin Zheng, Hao Ma, Michael R. Lyu, and Irwin King. Wsrec: A collaborative filtering based web service recommender system. In *Proceedings of the 7th IEEE International Conference on Web Services*, 2009.

[255] Zibin Zheng, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Ftcloud: A ranking-based framework for fault tolerant cloud applications. In *Proceedings of the 21st IEEE International Symposium on Software Reliability Engineering*, pages 398–407, San Jose, California, USA, 2010.

[256] Zibin Zheng, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Component ranking for fault-tolerant cloud applications. *IEEE Transactions on Service Computing*, 2011.

[257] Chao Zhou, Guang Qiu, Kangmiao Liu, Jiajun Bu, Mingcheng Qu, and Chun Chen. SOPING: A Chinese Customer Review Mining System. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 741–742, Singapore, 2008.

[258] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–328, 2004.

[259] Tom Chao Zhou and Irwin King. Automobile, car and bmw: Horizontal and hierarchical approach in social tagging systems. In *Proceedings of the 2nd ACM Workshop on Social Web Search and Mining, in conjunction with CIKM*, pages 25–32, Hong Kong, 2009.

[260] Tom Chao Zhou, Chin-Yew Lin, Irwin King, Michael R. Lyu, Young-In Song, and Yunbo Cao. Learning to suggest questions in online forums. In *Proceedings of AAAI*, pages 1298–1303, 2011.

[261] Tom Chao Zhou, Michael R. Lyu, and Irwin King. A classification-based approach to question routing in community question answering. In *Proceedings of CQA*, 2012.

[262] Tom Chao Zhou, Hao Ma, Irwin King, and Michael R. Lyu. Tagrec: Leveraging tagging wisdom for recommendation. In *Proceedings of IEEE International Symposium on Social Intelligence and Networking*, 2009.

[263] Tom Chao Zhou, Hao Ma, Michael R. Lyu, and Irwin King. Userrec: A user recommendation framework in social tagging systems. In *Proceedings of AAAI*, pages 1486–1491, 2010.

[264] Tom Chao Zhou, Xiance Si, Edward Y. Chang, Irwin King, and Michael R. Lyu. A data-driven approach to question subjectivity identification in community question answering. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada, 2012.

[265] S. Zhu, K. Yu, and Y. Gong. Stochastic relational models for large-scale dyadic data using mcmc. *Advances in Neural Information Processing Systems*, 21:1993–2000, 2009.

[266] X. Zhu. Semi-supervised learning literature survey. 2005.

[267] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[268] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, page 912C919, 2003.

[269] X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1052–1059. ACM, 2005.

[270] Y. Zhu, J.X. Yu, H. Cheng, and L. Qin. Graph classification: a diversified discriminative feature selection approach. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 205–214. ACM, 2012.

[271] C.N. Ziegler and G. Lausen. Analyzing correlation between trust and user similarity in online communities. *Trust Management*, pages 251–265, 2004.

[272] J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys*, 38(2):6, 2006.